A PARAMETER SURVEY OF A 2-D HOMOGENEOUS AND HETEROGENEOUS SIMULATION OF A TRIGA REACTOR USING AGENT WITH A REFLECTIVE AND VACUUM BOUNDARY CONDITION IN ISOTROPIC SCATTERING

by

Micah Kingston

A thesis submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Master of Science

in

Nuclear Engineering

Department of Civil and Environmental Engineering

The University of Utah

August 2011

The University of Utah Graduate School

STATEMENT OF THESIS APPROVAL

The thesis of

Micah Kingston

has been approved by the following supervisory committee members:

Tatjana Jevremovic, Chair 5/31/2011
Date Approved

Dong-Ok Choe, Chair 5/31/2011
Date Approved

Siva Guruswamy, Chair 5/31/2011
Date Approved

and by Paul Tikalsky, Chair of

the Department of Civil and Environmental Engineering

and by Charles A. Wight, Dean of the Graduate School.

ABSTRACT

The University of Utah TRIGA reactor (UUTR) was surveyed and simulated using a deterministic neutron transport code known as Arbitrary Geometry Neutron Transport, or AGENT. The UUTR was simulated using a homogeneous test material as well as a heterogeneous set of much more realistic cross-sections (provided by a former graduate student at the University of Utah) that are similar to the actual material used in the UUTR. The survey was done with an emphasis on a reflective boundary condition resulting in a k-infinite value, and a vacuum boundary condition resulting in a k-effective value to find the optimal Method of Characteristics (MOC) parameters. The four MOC parameters that were surveyed were the polar angle scheme and number, the number of azimuthal angles, the ray separation, and the number of edges per segment. The optimized values for these parameters were found to be 1 and 2 for the polar angle scheme and number, 36 azimuthal angles, a ray separation of 0.02 cm, and 4 edges per segment.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Tatjana Jevremovic, for her guidance during my studies under her. I especially would like to thank her for providing a path for me to finish the degree I started and getting me to the next step in my professional life.

I would also like to thank Dr. Dong-Ok Choe for help and teaching throughout my graduate career and for being on my supervisory committee.

I would also like to thank Dr. Siva Guruswamy for the time he has taken on my behalf and for being on my supervisory committee.

I especially wish to thank my family, especially my mother and father, for their faith in me, and my wife for her never-ending patience and love and support throughout my education.

CHAPTER I

INTRODUCTION

The University of Utah has housed a TRIGA reactor for over three decades now. The University of Utah TRIGA Reactor (UUTR) is designed to be able to safely operate at a power level of up to 1 MW. However, the UUTR is currently licensed to operate at 100 kW with a pending upgrade and relicensing to 250 kW.

The TRIGA reactor is the most common research reactor in America today with a total of 35 reactors within the United States and several more abroad, making a total of 66 total TRIGA reactors built in the world. The first TRIGA reactor was commissioned 1958, with many that followed and many of the originally installed TRIGAs are in operation today.

The TRIGA reactor can be used in a variety of applications. Some of which are non-destructive testing, production of radioisotopes (used in medicine as well as industry), nuclear research, and often the most popular at university level, to train graduate students on reactor operation and maintanence.

Motivation

The goal of this report is to perform a survey of the UUTR in 2-D using Arbitrary Geometry Neutron Transport (AGENT) to discover the most effective Method of Characteristics (MOC) parameters to be used to maximize the accuracy of the model of the UUTR in AGENT. The four MOC parameters that will be surveyed are the polar angle scheme and number, the number of azimuthal angles, the ray separation, and the number of edges used per surface.

Scope of Work

- The first survey that will be performed on the UUTR is a survey using a homogeneous test material. While performing this survey, the four MOC parameters will be optimized to find their effect on k-infinite (for the reflective boundary condition) and k-effective (for the vacuum boundary condition).
- The second survey that will be performed on the UUTR is a survey using a heterogeneous test material. There will also be an expanded survey on the CACTUS polar angle scheme for the heterogeneous case. While performing this survey, the four MOC parameters will be optimized to find their effect on k-infinite (for the reflective boundary condition) and k-effective (for the vacuum boundary condition).

Outline

The first chapter is the introduction. The second chapter is an overview of the method of characteristics in general and then it is explained how the MOC is used in

terms of neutron transport in AGENT. The third chapter shows the methods used to model the UUTR, especially with regards to the AGENT input file. The fourth chapter discusses the results of the survey of the homogeneous UUTR. The fifth chapter discusses the results of the survey of the heterogeneous UUTR. The sixth chapter talks of conclusions and recommendations for future work.

CHAPTER II

METHOD OF CHARACTERISTICS IN SOLVING THE NEUTRON TRANSPORT EQUATION

General Case of MOC

In general, the method of characteristics (MOC) can be used to simplify a system of algebraic or differential equations. In some cases, the MOC can be used to reduce the number of variables in an equation to simplify it; in other cases, the MOC can be used to change a partial differential equation (PDE) to a simpler ordinary differential equation (ODE). An example of the simplest case is in the following demonstration.

One can use MOC to represent both X and Y as S. To do this, one starts with equations 2.1a and 2.1b:

$$\frac{dX}{dS} = \cos(\theta) \tag{2.1a}$$

$$\frac{dY}{dS} = \sin(\theta) \tag{2.1b}$$

First, get the Xs and Ys on the left side and the Ss on the right side, as seen in equations 2.2a and 2.2b.

$$dX = \cos(\theta)\, dS \tag{2.2a}$$

$$dY = \sin(\theta) dS \tag{2.2b}$$

Then, take the integral of both sides as displayed in equations 2.3a and 2.3b. Then simplify as shown in equations 2.4a and 2.4b.

$$\int dX = \int \cos(\theta)\, dS \tag{2.3a}$$

$$\int dY = \int \sin(\theta) dS \tag{2.3b}$$

$$X = S\cos(\theta) \tag{2.4a}$$

$$Y = S\sin(\theta) \tag{2.4b}$$

Now, square both equations on both sides and add the left-hand sides together and the right-hand side of both equations as seen in equation 2.5.

$$X^2 + Y^2 = S^2[\cos(\theta)^2 + \sin(\theta)^2] \tag{2.5}$$

But of course, the well-known trigonometric function sets the sinusoids equal to one, leaving only equation 2.6.

$$X^2 + Y^2 = S^2 \tag{2.6}$$

Using the MOC, one can represent X and Y by only using S, which can significantly simplify the equation. This can be shown graphically by Figure 2.1 realizing that equation 2.6 is actually the Pythagorean Theorem.

MOC Used in Isotropic Neutron Transport

Now, one can apply the MOC to the Boltzmann neutron transport equation to change the complicated PDE to a much simpler ODE as is accomplished in AGENT. To do this, start with the Boltzmann Transport Equation as seen in equation 2.7 [2], [3],[4].

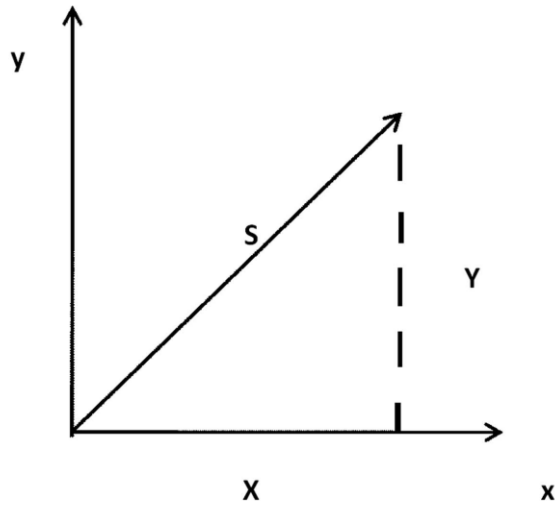


Figure 2.1. Physical Interpretation of S in Terms of X and Y.

$$\hat{\Omega} \cdot \nabla \Psi_g(\bar{r},\hat{\Omega}) + \Sigma_g(\bar{r})\Psi_g(\bar{r},\hat{\Omega}) = Q_g(\bar{r}) \quad (2.7)$$

where $\Psi_g(\bar{r},\hat{\Omega})$ = Angular flux at vector $\bar{r}$ in direction of the solid angle $\hat{\Omega}$ in energy group g.

$\Sigma_g(\bar{r})$ = Total cross-section at vector $\bar{r}$ in energy group g.

$Q_g(\bar{r})$ = Isotropic source term at vector $\bar{r}$ in energy group g.

It is also important to note that the isotropic source term can be broken up into two parts, the isotropic scattering portion, and the fission portion, as described by equation 2.8.

$$Q_g(\bar{r}) = Siso_g(\bar{r}) + F_g(\bar{r}) \quad (2.8)$$

The isotropic scattering source term and the fission source term are defined by equations 2.9 and 2.10

.

$$Siso_g(\bar{r}) = \frac{1}{4\pi}\sum_{g'} \Sigma_{iso,g'\rightarrow g}(\bar{r})\phi_{g'}(\bar{r}) \quad (2.9)$$

$$F_g(\bar{r}) = \frac{\chi_g}{4\pi k_\infty}\sum_{g'}^{g} \nu\Sigma_{f,g'}(\bar{r})\phi_{g'}(\bar{r}) \quad (2.10)$$

where $\Sigma_{iso,g'\rightarrow g}(\bar{r})$ = Isotropic scattering cross-section at vector $\bar{r}$ from energy group g' to g.

$\phi_{g'}(\bar{r})$ = Scalar flux at vector $\bar{r}$ in energy group g'.

$\nu\Sigma_{f,g'}(\bar{r})$ = Average number of neutrons per fission multiplied by the fission cross section at vector $\bar{r}$ in energy group g'.

χ_g = Fission spectrum in group g.

k_∞ = Infinite multiplication factor (assumes no neutron leakage). For a more thorough discussion of the multiplication factor refer to Appendix A.

To perform the MOC on the isotropic Boltzmann neutron transport equation, the first term in equation 2.7 is expanded to show the x and y components and is rewritten to become equation 2.11:

$$\hat{\Omega}\cdot\left[\frac{\partial\Psi_g}{\partial x}\hat{i}+\frac{\partial\Psi_g}{\partial y}\hat{j}\right]+\Sigma_g(\bar{r})\Psi_g(\bar{r},\hat{\Omega})=Q_g(\bar{r}) \tag{2.11}$$

Now, using the MOC, one can introduce S, which is a characteristic ray that is a straight line in the direction of the solid angle $\hat{\Omega}$. This is shown in equations 2.12a and 2.12b.

$$\frac{\partial x}{\partial s}=cos\theta \tag{2.12 a}$$

$$\frac{\partial y}{\partial s}=sin\theta \tag{2.12 b}$$

Just as was shown in the simple example above, the 2D plane in x and y can now be represented in the parametric plane S, so the partial differential equation that is the Boltzmann neutron transport equation becomes a much simpler ordinary differential equation as seen in equation 2.13 below.

$$\frac{d\Psi_g(\bar{r},\hat{\Omega})}{dS} + \Sigma_g(\bar{r})\Psi_g(\bar{r},\hat{\Omega}) = Q_g(\bar{r}) \tag{2.13}$$

But this parametric plane S must be accounted for and tracked within whatever geometry is to be simulated in AGENT. This is accomplished by three additional parameters to discretize both the spatial and angular environment modeled in AGENT. The first parameter addresses the spatial environment, which is split up into zones with a constant flux (identified by index i). To take care of the angular environment, the next two parameters are parallel characteristic rays (identified by index k) in conjunction with varying azimuthal rays (identified by index m). This is demonstrated visually by Figure 2.2 [2,3,4].

With the first term of the Boltzmann's neutron transport equation discretized, equation 2.13 now becomes equation 2.14.

$$\frac{d\Psi_{g,m,i,k}}{dS_{m,i,k}} + \Sigma_{g,i}\Psi_{g,m,i,k} = Q_{g,i} \tag{2.14}$$

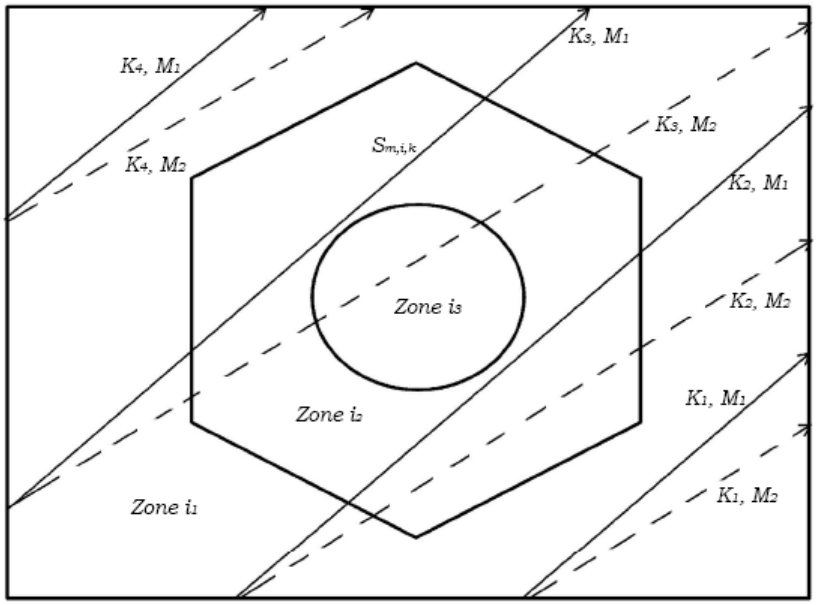


Figure 2.2. Simulated Geometry Discretized by the MOC Parameters Used in AGENT.

Breaking up the discretized source term now produces equation 2.15.

$$Q_{g,i} = Siso_{g,i} + F_{g,i} \tag{2.15}$$

The isotropic scattering source term and the fission source term now become what is shown in equations 2.16 and 2.17.

$$Siso_{g,i} = \frac{1}{4\pi}\sum_{g'} \Sigma_{iso,g'\rightarrow g,i}\phi_{g',i} \tag{2.16}$$

$$F_{g,i} = \frac{\chi_{g,i}}{4\pi\, k_{\infty}}\sum_{g'} \nu\Sigma_{f,g',i}\,\phi_{g',i} \tag{2.17}$$

where

$$\phi_{g',i} = \sum_m \omega_m \Psi_{g.m.i} \qquad (2.18\ a)$$

and ω_n is the angular weight of angle m, as seen in equation 2.18b.

$$\sum_n \omega_n = 1 \qquad (2.18\ b)$$

From there, it is now possible to solve for the average angular flux in group g, direction m, and zone i. This is accomplished by equation 2.19.

$$\overline{\Psi_{g,m,\imath}} = \frac{\sum_k \overline{\Psi_{g,m,\imath,k}} S_{m,i,k} \delta A}{\sum_k S_{m,i,k} \delta A} \qquad (2.19)$$

where $\overline{\Psi_{g,m,\imath,k}}$ is found by the procedure used in equations 2.20 through 2.25.

$$\overline{\Psi_{g,m,\imath,k}} = \frac{1}{S_{m,i,k}} \int_0^{S_{m,i,k}} \Psi_{g,m,i,k}(s) ds \qquad (2.20)$$

By putting the angular flux into equation 2.20, the equation 2.21 is the result.

$$\overline{\Psi_{g,m,\imath,k}} = \frac{1}{S_{m,i,k}} \left[\Psi_{\mathrm{g,m,i,k}}(0) + \mathrm{Q_{g,i}} S_{m,i,k} - \Psi_{\mathrm{g,m,i,k}}\left(S_{m,i,k}\right) \right] \qquad (2.21)$$

This simplifies to equation 2.22.

$$\overline{\Psi_{g,m,i,k}} = \frac{\Delta_{g,m,i,k}}{S_{m,i,k}\Sigma_{g,i}} + \frac{Q_{g,i}}{\Sigma_{g,i}} \tag{2.22}$$

where

$$\Delta_{g,m,i,k} = \Psi_{g,m,i,k}(0) - \Psi_{g,m,i,k}(S_{m,i,k}) \tag{2.23}$$

Finally, this leads to equation 2.24 for the average angular flux in group g, direction m, and zone i.

$$\overline{\Psi_{g,m,i,k}} = \frac{\sum_k \left(\frac{\Delta_{g,m,i,k}}{S_{m,i,k}\Sigma_{g,i}} + \frac{Q_{g,i}}{\Sigma_{g,i}}\right) S_{m,i,k}\delta A}{\sum_k S_{m,i,k}\delta A} \tag{2.24}$$

This implies that the scalar flux equals equation 2.25.

$$\phi_{g,i} = \sum_m \omega_m \overline{\Psi_{g,m,i}} \tag{2.25}$$

Simplifying, we get the equation 2.26 for the scalar flux for energy group g, and zone i.

$$\phi_{g,i} = \frac{Q_{g,i}}{\Sigma_{g,i}} + \frac{1}{\Sigma_{g,i}V_i} \sum_m \omega_m \sum_k \Delta_{g,m,i,k}\delta A \tag{2.26}$$

where V_i = Volume for zone i.

With the appropriate boundary conditions and the neutron source initial conditions, AGENT runs a continuous iteration that repeats the solving of the previous equations until the convergence criteria are met for both the multiplication factor and the flux. These criteria are specified in the input file that will be discussed in the next chapter.

Brief Review of MOC Parameters in AGENT

When performing the survey of the MOC parameters in AGENT, it is important to be able to visualize what is actually being changed. The four MOC parameters that were surveyed were the polar angle scheme and number, the azimuthal angles, the ray separations, and the number of edges per boundary.

Referring to Figure 2.2 once more, it can be seen that the k's are the polar angles and the m's are the azimuhal angles. The ray separation would be the distance between each ray as seen in Figure 2.2. Finally, for the number of boundary edges, it is necessary that no edge is shared between two different zones, and that there must be at least two rays on each edge [2].

CHAPTER III

METHODS USED TO MODEL THE UNIVERSITY OF UTAH TRIGA REACTOR USING AGENT

In order to perform the analysis of the UUTR in AGENT, it was necessary to follow the programming conventions established within AGENT for the input file. The MOC parameters were varied to complete a 2-D survey of the TRIGA reactor first using a test material for the entire reactor to establish the proper geometries, known as the homogeneous UUTR. Then, the materials were changed to reflect the actual condition of the TRIGA reactor, known as the heterogenous UUTR. All of the MOC parameters are located within the spatial resolution portion of the input file, or the $MAP card. All of the conventions used to simulate the UUTR will be reviewed as well as the survey that was performed within the $MAP to optimize the spatial resolution of the UUTR simulation.

AGENT Input File

To correctly fill out the input file for AGENT, there are six main sections that must be covered. These are the title and calculation mode or $TITLE card, the geometry modeling or the $GEOM card, the material modeling or the $XSEC card, the calculation accuracy or $OPT card, output setting or $EDIT card, and the spatial resolution or $MAP

card which is where the MOC parameters can be edited. Each of these sections will be briefly discussed, especially with regards to how they pertained to the modeling of the UUTR.

$Title

The first section within any input file for an AGENT program is the $TITLE card. This portion is meant for the user to define the title of the file that will be shown in the output file as the AGENT executable is run, and to specify the calculation mode to be used. In AGENT, there are three possible calculation modes: the Multi-mode (used for full-core calculations which utilize MULTI-AGENT), the Arbitrary mode (used for arbitrary complex domains), and the Lattice mode (used for lattice domains or repeated geometries that can be fit into an array) [3]. Because the UUTR can be described by a hexagonal array of fuel, water, and control rods, the Lattice mode was used for the UUTR.

Geometry: The $GEOM Card

The geometry of the UUTR is a relatively simple hexagonal lattice of fuel, control rods, and other materials. To build the geometry of the UUTR, it is first necessary to build the primitives of which the UUTR is composed. Then, composites and volumes can be defined and combined in the appropriate lattice and lattice geometry to construct the entire UUTR. The UUTR is located in the Merrill Engineering Building in the University of Utah. A layout similar to the UUTR can be seen in Figure 3.1.

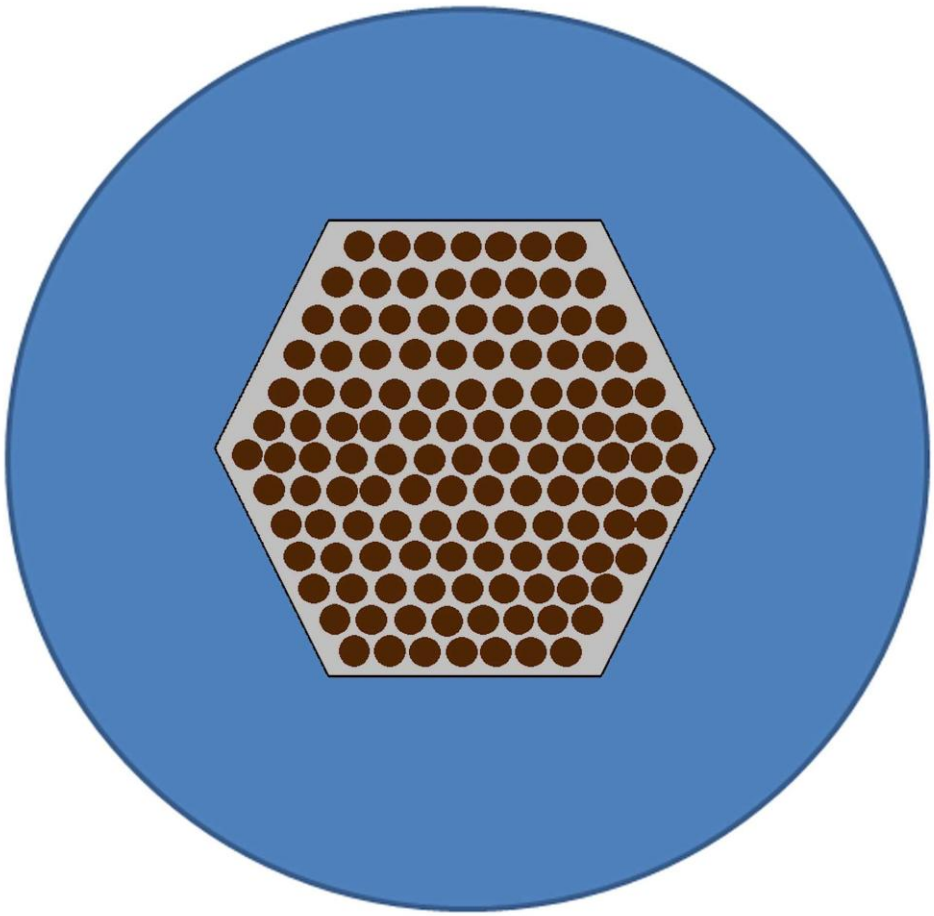


Figure 3.1. Layout Similar to the University of Utah TRIGA Reactor.

To define these primitives, it is necessary to use the convention required for each cell to be modeled. It is also important to note that AGENT can automatically fill in a lattice of repeating geometries if applicable. So, for the UUTR, each cell that is either a unique size or an orientation of one of the primitive shapes within the lattice of the UUTR must be defined. There are four basic shapes modeled at various orientations that are used as primitives in the modeling of the UUTR: the pentagon, the half hexagon, the hexagon, and the cylinder. For example, the first cell to be defined will be the bottom left corner of the water that is a pentagon which is turned on its top left side. The programming conventions for all orientations of a hexagon can be seen in Figure 3.2.

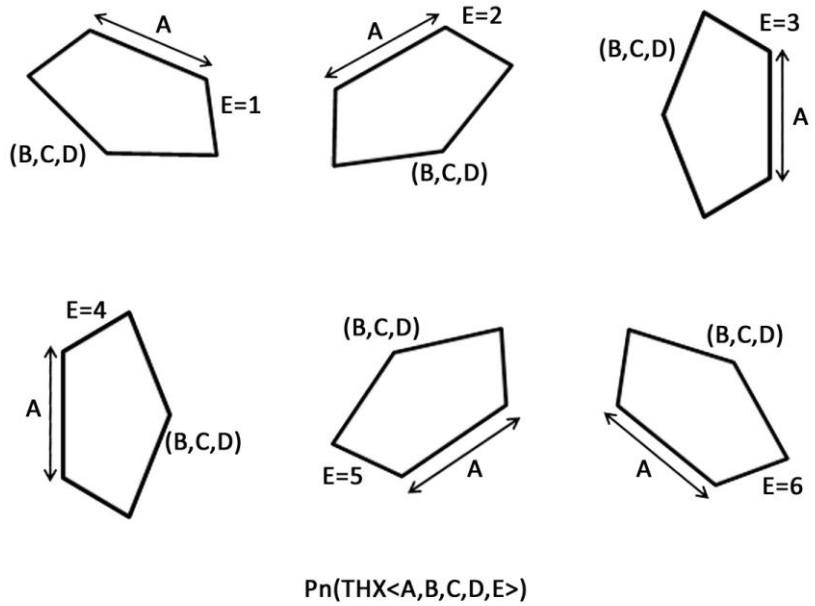
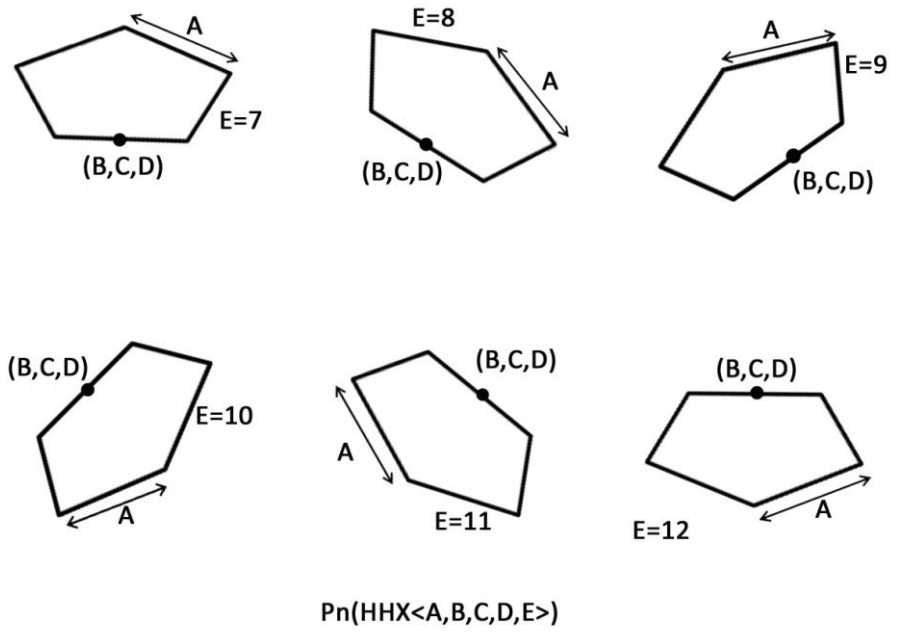


Figure 3.2. Conventions Used for Programming a Pentagon in AGENT.

The Pn stands for the cell (where n is the cell #). The orientation that was used in the first cell was 1, so E = 1. The rest of the coding for the first primitive followed the format seen in Figure 3.2 where A is the length of the side of the pentagon and B, C, and D correspond to the X, Y, and Z coordinates of the top point of the pentagon [3].

The coding convention that is used for the half hexagon is similar to the pentagon in that A is the length of the side of the hexagon and B, C, and D are the coordinates of the center of one of the sides of the hexagon. Finally, E also corresponds to the orientation of the half hexagon. Figure 3.3 shows an example for each possible orientation of the half hexagon [3].



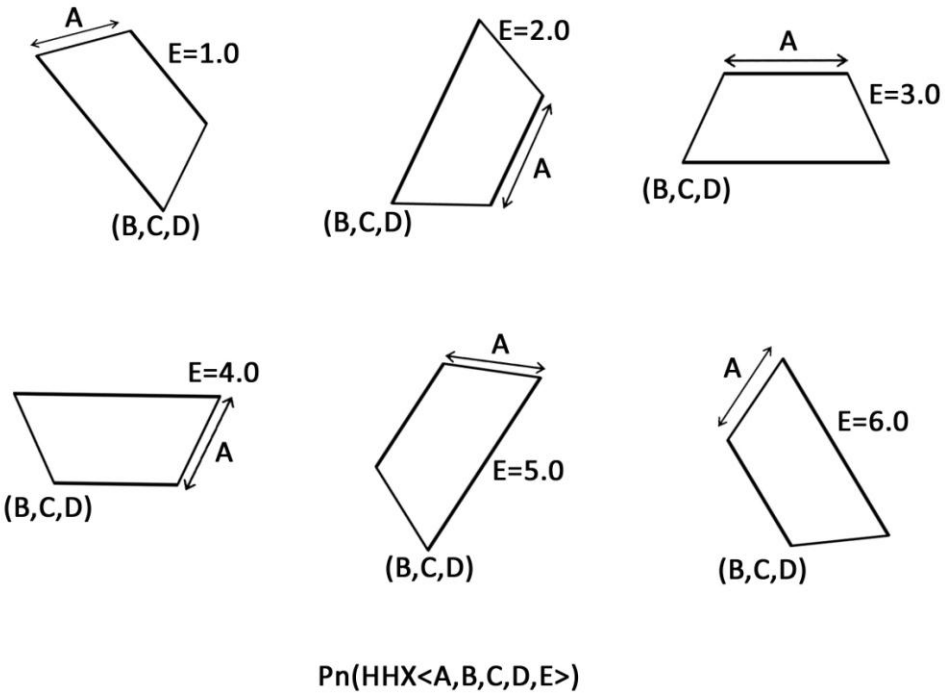
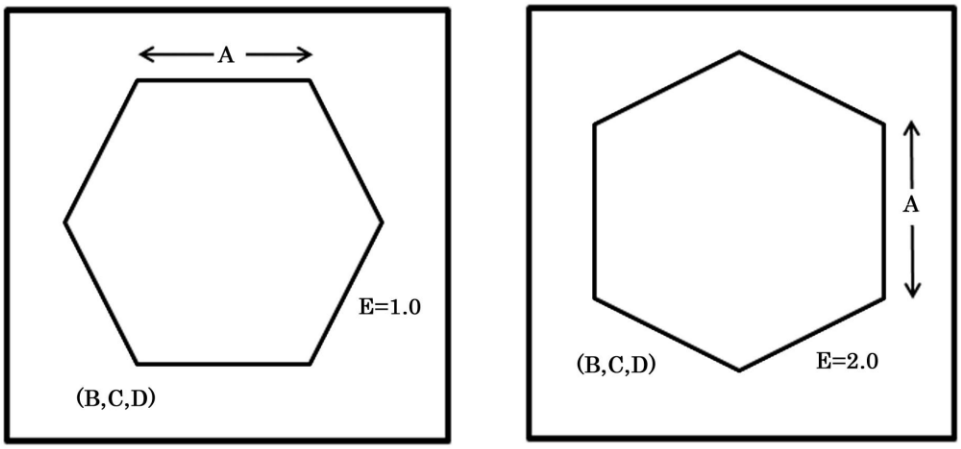


Figure 3.3 Half Hexagon Orientation Examples in AGENT.

For the hexagon, the input file programming convention is very similar to the pentagon and the half hexagon. The A defines the width of a side, the B, C, and D defines the coordinates of the bottom left corner of the hexagon, and the E defines the orientation of the hexagon. There are two possible orientations: standing on a point or on a flat edge. This is shown in Figure 3.4.

The last primitive that was used in modeling the UUTR was the cylinder. The cylinder was used to represent the fuel that was inside the various hexagons throughout the UUTR. Multiple cylinders needed to be placed in each hexagon to distinguish between the cladding and the fuel, as well as making the calculations performed by AGENT more precise (this is accomplished by adding additional smaller geometries). It is possible to input the number of equidistant subdivisions in the cylinder but that is not what was done in modeling the UUTR because the distance between each "ring" within the model is not equal. An example of the cylinder coding can be seen in Figure 3.5.



Pn (HEX<A,B,C,D,E>)

Figure 3.4. The Hexagon Programming Input in AGENT.

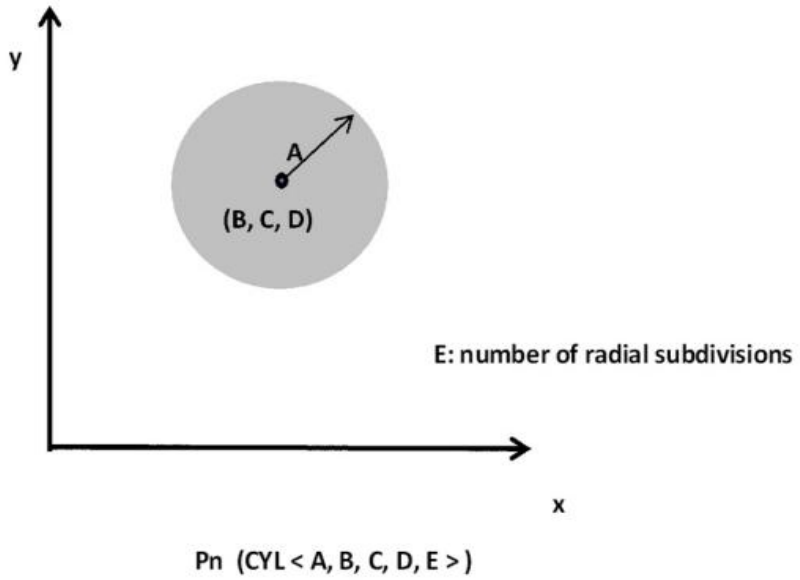


Figure 3.5. Cylinder Programming Input in AGENT.

R-Functions

AGENT handles the geometry of the UUTR beautifully. AGENT uses the principle of R-functions to combine or manipulate predefined primitive. An R-function can perform 3 basic operations. The first operation is an "and" operator which will build a composite of two primitives overlapping each other. The second operation is a "union" or an "or" operator which builds a single composite of all the combined primitives. The third operation the R-function performs is the "minus" operator. This subtracts the second primitive from the first and builds a composite from whatever is left. Using the R-function allows AGENT to go from simple primitives all the way up to a reactor core.

To make the necessary composites for the UUTR, each primitive that is defined needs to be entered into the input file as a composite as well. For example, a primitive that is represented as "P10" would create the corresponding composite as "C10 (10)," which means the composite labeled "C10" is composed of the primitive "P10." Any

additional composites that were created through the use of the R-function as previously described used the following syntax: "C50 (50 – 51)," which reads to mean the composite #50 equals the difference of primitive 50 – primitive 51.

After all the composites are defined in the input file, the next step is to input the volumes for the UUTR. This is done by a similar method as the composites. To follow along the same example as the composite but for the volumes, a volume representing the "C10" composite is represented by "V10 = (P10)#."

Geometry: The $LATT Cards

The next portion of the input file that is necessary to model the geometry of the UUTR is the $LATT Card. There were four different types of $LATT cards used in modeling the UUTR. The first type was the $LATT card. The $LATT card covers the types of geometries for each unit cell that are to be used throughout the entire UUTR. For example, one unit cell might be on the bottom corner of the UUTR of which there is only one and another cell might have multiple rings or geometries within its cell and might be used throughout the reactor model. In the homogeneous model of the UUTR, there were 13 separate lattices required whereas in the heterogeneous model there were 15.

The second type of $LATT card that was used to model the UUTR is the %LATG card. The %LATG card is used to indicate the location or the "lattice map" of each unit cell with respect to one another. This also allows the use of multiple copies of the same type of unit cell to be used throughout the lattice which is utilized in the modeling of the

UUTR. The %LATG card also defines the specific shape that the collection of unit cells are placed inside; in the case of the UUTR, a hexagon was used [3].

The third type of $LATT that was used in modeling the UUTR is the $LATS card. The $LATS card takes into account the nature of the MOC solution by making each portion of a cell having the same flux throughout. Because of this, each unit cell must be broken into smaller portions to increase the accuracy of the model. It is important to note that as one increases the unit cell portions and corresponding accuracy of the model, the iteration time required to build the model is also increased. In the case of the UUTR, it was necessary to use "2 2 4" for all of the unit cells that were of a single geometry and material, and "2 2 3" was used for the unit cells that were of multiple geometries and materials (it was allowable to use the 3 rather than the 4 because there were already reduced portion sizes within these cells) [3].

The fourth and final type of $LATT card that was used in modeling the UUTR was the %LATM card. The %LATM card is similar to the %LATG, except that it focused on the material type of each unit cell within the lattice rather than its geometry. This is done within the $XSEC card.

Material: $XSEC Card

The $XSEC card is the portion of the input file that is used to define the materials used in modeling the UUTR. The first part of the $XSEC card sets up the defaults to be expected throughout the material card and is described by 5 digits. The first digit indicates the number of energy groups that are used to describe every material simulated in the input file. The next digit indicates the first energy group that has in-scattering.

The third digit is the number of energy groups that are below 0.625 eV (thermal energy), and the fourth digit indicates which group is the first group to pass this energy threshold. The energy groups within AGENT are described from the highest energy to the lowest energy. The last digit is a zero by default [3].

After the first line is setup within the $XSEC card, the limits of the energy groups are setup for the simulation and the %LATM card is implemented as described earlier. The next part of the material card is to define each material type by a single material or a group of materials. It is necessary to input the multigroup macroscopic cross-sections for every material definition. This can get very lengthy as the number of energy groups representing the material increases. There were 4 groups used for the homogeneous case and 7 groups used to describe the heterogeneous model of the UUTR. For the survey of both the homogeneous and heterogeneous models of the UUTR, the material cross-section data were provided to me by a former graduate student at the University of Utah.

Accuracy of Calculations: The $OPT Card

Because AGENT uses an iteration process to solve the neutron transport code, it becomes beneficial to allow the user to adjust the accuracy of the iteration keeping in mind that the more accurate the calculation, the longer the iteration will take. AGENT uses two ways to determine that the iteration has converged. The first way is to check what the relative difference is between the new multiplication factor and the old one. This is the absolute value of the new multiplication factor minus the previous multiplication factor all over the previous multiplication factor. This is the first convergent criterion and is referred to as kdiff.

The second method that is used to check if the iteration has converged is to check the maximum relative difference of the zone fluxes throughout all of the zones. This is known as the flux-diff-max term. If the kdiff term is less than the first value the user enters in this section of the input file (known as epsilon1), and if the flux-diff-max term is less than the second term the user inputs (known as epsilon2), then the iteration is said to converge and the iterations stop. The values for epsilon1 and epsilon2 for the homogeneous option were 0.001 and 0.01; and the values used for the heterogeneous case were 0.00001 and 0.0001, respectively [3].

The next part of the $OPT card are the two boundary options. The first boundary option is used to describe the outer-most primitive of the entire geometry. In this case, it is the hexagon that holds the entire lattice geometry of the core. This is described by P140 so naturally, this input was 140 for both the homogeneous and heterogeneous cases. The second boundary option is a description of the number and type of outer bodies. Because there were no elements outside the hexagon, the default became "0 1" for both the homogeneous and heterogeneous cases [3].

The remaining options within the $OPT card is the "Volume Reconstruction" tag, the "Angular Flux Transfer Option," the "CMFD Option," and the "Ray Tracing Option," respectively [3]. For the Volume tag, option 2 was used for both the homogeneous and the heterogeneous which is used to allow the user to specify the zone volumes within the input file. For the angular flux card, 0 was used for both the homogeneous and heterogeneous cases, which means that for any flux that crosses an edge of any geometry within the simulation, the flux is calculated as an average. For the CMFD option, 0 was used for both the homogeneous and heterogeneous cases, which disables the CMFD

option for the input file. Finally, the ray tracing option was 0 for both the homogeneous and heterogeneous cases, which means that the ray tracing that AGENT performs will be independent of the geometry that is in the model.

Output: The $EDIT Card

The $EDIT card is used to allow the user to define exactly what is to be present in the output file. The mode that was used in both the homogeneous case and the heterogeneous case was: "ON 0 0"; which is the default to allow all of the information from the iteration to be printed in the output file [3].

Spatial Resolution: The $MAP Card

The $MAP card is a very important part of the AGENT input file as it holds all of the MOC parameters which are used to perform the survey of the UUTR as well as others. During the survey of the UUTR, four of these parameters were surveyed for both the reflective and vacuum boundary conditions. This was done for both the homogeneous and heterogeneous simulations of the UUTR. The four parameters that were surveyed were the polar angle scheme and number, the azimuthal angles, the ray separation, and the number of edges per side. To perform the survey with multiple parameters, each survey was performed until the optimum parameter was found and held constant as the remaining surveys were completed. Each parameter, whether it was surveyed or not, will be discussed. There will also be a discussion of the boundary conditions that can be simulated in AGENT along with the boundary conditions that were used in the survey of the homogeneous and heterogeneous models of the UUTR.

The first parameter that was surveyed in AGENT was the polar angle scheme and the number of polar angles used. There are two options for the polar angle scheme; the first option is the "Leonard and McDaniel" scheme (Leonard and McDaniel, 1995). The second option is the "CACTUS" quadrature scheme. The UUTR was surveyed for both options and the number of polar angles was varied from one to two for each option surveyed for the homogenous case, but the number of polar angles surveyed for the heterogenous case of the CACTUS option was from 1 to 6.

The second parameter of the UUTR that was surveyed was the number, range, and mode of the azimuthal angles used. The number of azimuthal angles used in the survey varied from twelve to thirty-six in multiples of four. The range of the azimuthal angle always remained at 180° throughout the survey. The mode also remained at 1 as both the homogenous and heterogeneous simulations of the UUTR were performed. The next item in the input file that also remained at a default value was the plot switch and material level; of which the default values were "0 1 0 1" [3].

The third parameter of the UUTR that was surveyed was the ray separations and the number of angles to be used for fast and thermal groups. Decreasing the ray separations increases the accuracy but can increase the iteration time substantially. During the survey, the ray separations were varied from 0.02 cm to 0.08 cm for both the homogeneous and heterogeneous cases of the UUTR. It was necessary to keep the number of angles for the fast and thermal groups at exactly one – half of the number of azimuthal angles throughout the survey. This means that half of the azimuthal angles are used for the fast groups and half of the azimuthal angles are used for the thermal groups.

The last parameter of the UUTR that was surveyed was the number of boundary edges. As AGENT runs the iterations of the UUTR, every boundary of each geometry within the model of the UUTR is subdivided into a specified number of segments known as boundary edges. As one increases the number of boundary edges, the accuracy goes up but the iteration time can go up very quickly as well. A range from four to thirty-six edges was surveyed for both the homogeneous case and the heterogeneous case of the UUTR. After the boundary edges, the next two lines, which are the last two lines of the input file, are defaulted to be "0."

Boundary Conditions

Another important part that was covered in the input file was the various boundary conditions. There are four types of boundary conditions that can be used in AGENT: the reflective boundary condition, the periodic boundary condition, the white boundary condition, and the vacuum boundary condition. Although the full parameter survey was only performed on the reflective and vacuum boundary conditions, the UUTR was modeled in the homogeneous case to examine the effects of each boundary condition. Each of these boundary conditions will be briefly discussed.

The first boundary condition that was looked at was the reflective boundary condition (option 0 in the input file). The reflective boundary condition does not allow anything to escape the geometry being modeled. For example, if there is a given flux of neutrons of a particular angle that hit the boundary edge of a given geometry, that flux will be "reflected" back into the geometry at a mirrored angle. This is illustrated by Figure 3.6.

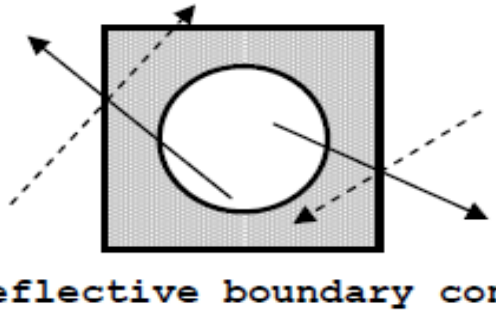


Figure 3.6. Illustration of the Reflective Boundary Condition.

The second boundary condition that was observed was the periodic boundary condition (option 1 in the input file). The periodic boundary condition causes whatever incoming flux on any particular edge of the given geometry to be set to the outgoing flux on the opposite edge of the incoming flux at the same angle. This creates the situation that throughout the geometry, the incoming flux from all angles equals the outgoing flux in all corresponding angles. This can be seen in Figure 3.7.

The third boundary condition that was observed was the "White" boundary condition (option 2 in the input file). The White boundary condition is such that all of the incoming angular fluxes at any particular boundary edge are set equal to a single value of a flux going outside of the given geometry so that the net flux across each edge of the given geometry is zero. An illustration of this is shown in Figure 3.8.

The last boundary condition that was observed was the vacuum boundary condition (option 4 in the input file). The vacuum boundary condition causes no reflection on the boundary edges of the given geometry. This means that any flux of neutrons that go outside the boundary of a given geometry never return. This can be seen in Figure 3.9.

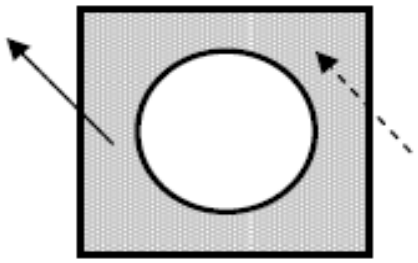


Figure 3.7. Illustration of the Periodic Boundary Condition.

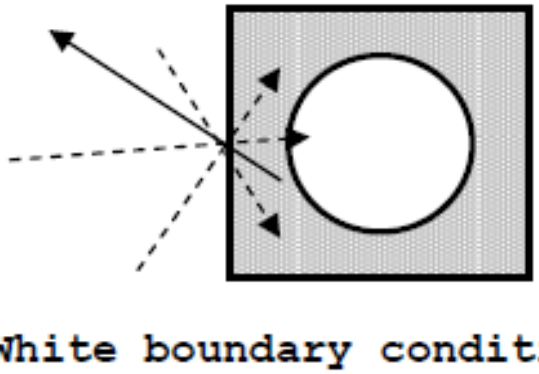


Figure 3.8. Illustration of the White Boundary Condition.

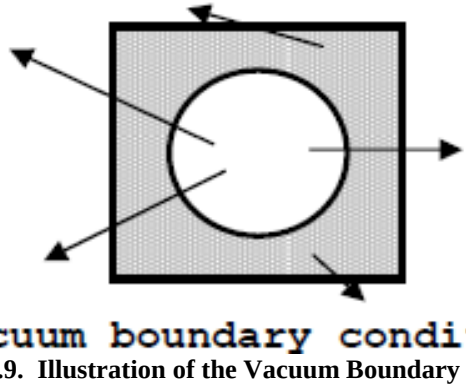


Figure 3.9. Illustration of the Vacuum Boundary Condition.

The full survey was performed in the homogeneous case and the heterogeneous case in both the reflective boundary condition, which simulates absolutely no leakage within the UUTR; and the vacuum boundary condition, which simulates not only leakage, but because there is no reflection whatsoever, it simulates slightly more leakage than the UUTR would actually produce. This is advantageous because it allows a conservative estimate of the neutron flux within the UUTR.

Differences in Modeling the Homogeneous UUTR and the Heterogeneous UUTR

The accuracy of the heterogeneous survey of the UUTR compared to the homogeneous survey of the UUTR increases substantially. This is because of the fact that instead of running a test material in the geometry of the UUTR, as is the case with the homogeneous UUTR, the material properties of what is actually in the physical UUTR were used. With the actual material being simulated, it was also necessary to increase the spatial resolution and additional rings were input into each cell within the geometry. Another upgrade that occurred from the homogeneous to the heterogeneous UUTR was that the simulations have gone from two energy groups to five. There was also a noticeable difference in the flux maps between the reflective and the vacuum boundary conditions, of which pictures in the proceeding chapters will be shown for each energy group as well as the absorption reaction rate, fission reaction rate, and the fission source.

CHAPTER IV

RESULTS OF THE SURVEY OF THE HOMOGENOUS UUTR

The UUTR was surveyed via the MOC parameters for the homogeneous and the heterogeneous cases in 2-D for the case of the reflective and the vacuum boundary conditions. There will be a brief discussion of the simulated flux maps due to the four different boundary condition options within AGENT. Then, the results of the survey of the four MOC parameters, namely the polar angle scheme, the azimuthal angles, the ray separation, and the number of edges per side for the vacuum and the reflective boundary condition for the homogeneous and heterogeneous simulations of the UUTR will be discussed. The simulations of the boundary conditions will be discussed first, then the homogeneous survey of the UUTR with the vacuum boundary condition, followed by the heterogeneous survey of the UUTR.

Boundary Conditions

In simulating the effects of the different boundary conditions, all four options were explored. These options include the reflective, periodic, white, and vacuum boundary conditions. For each case, the iteration was run in AGENT for a homogeneous

test material and then was input into a Matlab program to produce a visual image of the simulation. In each simulation, the first five scaled energy groups were produced as well as a scaled absorption reaction rate, a scaled fission reaction rate, and a scaled fission source. Because these terms are all scaled values, the visual effects of the boundary conditions are similar for each image produced. Therefore, only the group 1 scaler flux will be shown for each boundary condition.

Reflective Boundary Condition

The first boundary condition that was explored was the reflective boundary condition and can be seen in Figure 4.1. The reflective boundary condition does not allow anything to escape the geometry which would cause the scalar flux to be nearly uniform throughout the UUTR with the homogeneous test material as is observed in Figure 4.1.

Periodic Boundary Condition

The second boundary condition that was explored was the periodic boundary condition. The simulation of the group 1 scalar flux of the UUTR with the homogeneous test material for the periodic boundary condition can be observed in Figure 4.2.

The periodic boundary condition makes whatever incoming flux on any particular edge to be set to the outgoing flux on the opposite edge of the geometry at the same angle. It may not appear that there are several different geometries within the hexagon of the previous, but there are many because of the homogeneous test material. There are many hexagonal cells within the large hexagon of Figure 4.2 and due to the shape of the

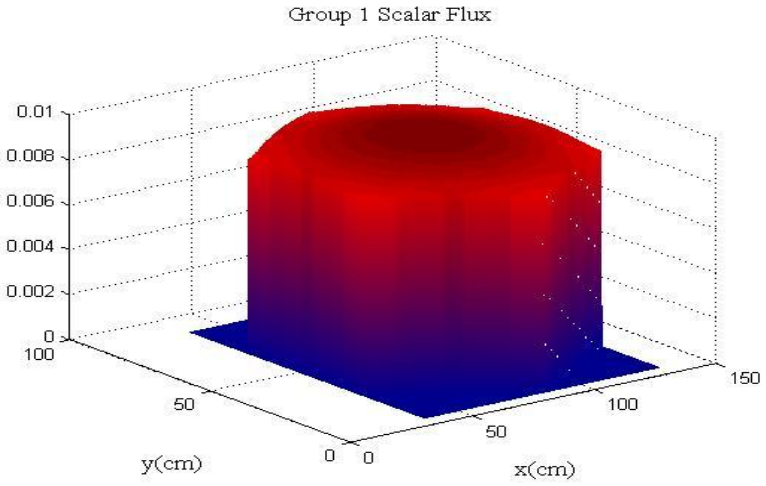


Figure 4.1. Group 1 Scaler Flux of the UUTR for the Reflective Boundary Condition.

hexagon and the nature of the periodic boundary condition, much of the fluxes tend to be at a maximum in the middle of the hexagon.

White Boundary Condition

The third boundary condition that was explored was the white boundary condition. The group 1 scalar flux of the white boundary condition can be seen in Figure 4.3.

The white boundary condition caused the scaler flux of the UUTR simulation to be essentially uniform with the homogeneous test material. This is because the white boundary condition causes the value of the flux at each geometrical boundary within the simulation to be zero, which is what was expected for the white boundary condition.

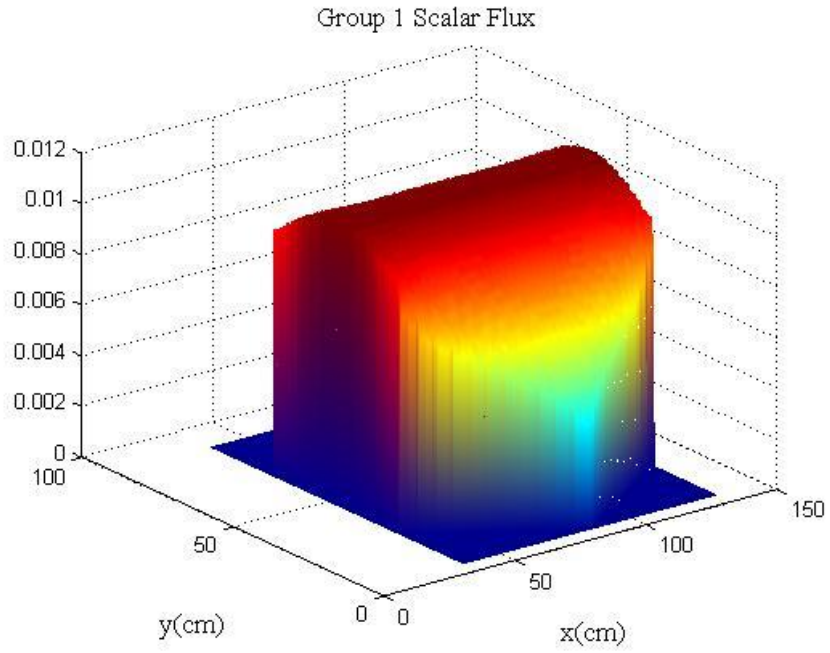


Figure 4.2. Group 1 Scaler Flux of the UUTR for the Periodic Boundary Condition.

Vacuum Boundary Condition

The fourth boundary condition that was explored was the vacuum boundary condition. The simulation of the group 1 scaler flux of the UUTR with the homogeneous test material for the vacuum boundary condition can be observed in Figure 4.4.

The group 1 scaler flux of the UUTR peaks in the center of the model because all of the neutron interactions that hit the edge of any geometry within the simulation are leaked out.

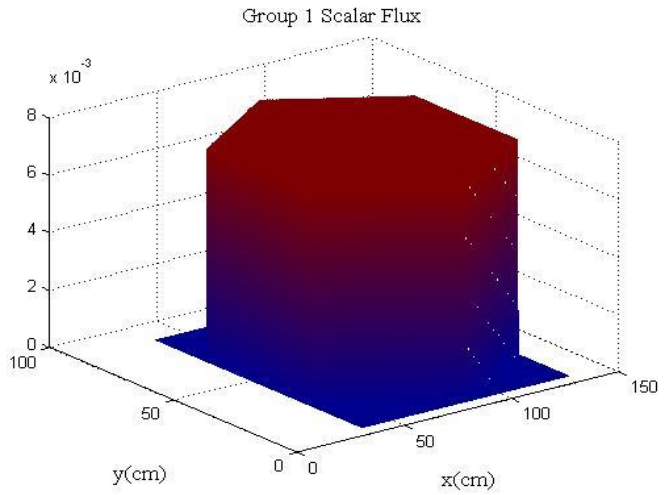


Figure 4.3. Group 1 Scaler Flux of the UUTR for the White Boundary Condition.

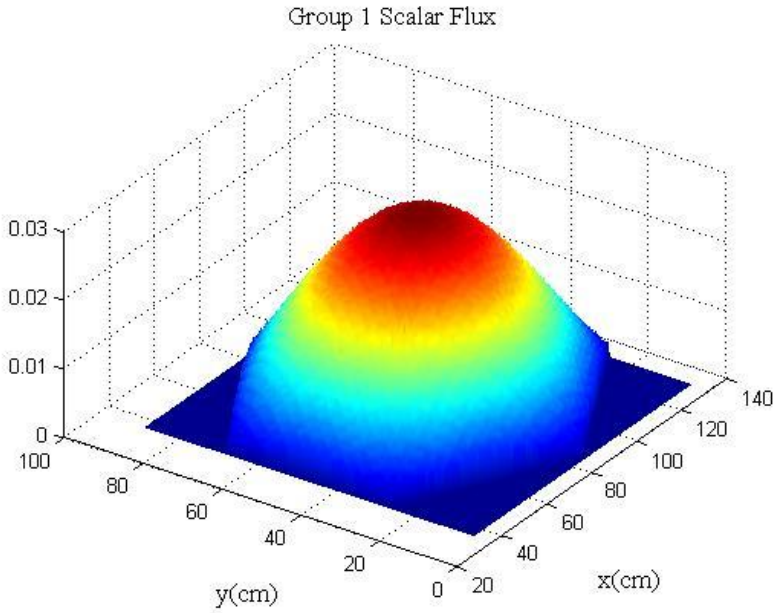


Figure 4.4. Group 1 Scaler Flux of the UUTR for the Vacuum Boundary Condition.

Survey of the UUTR with a Homogeneous Test Material

The UUTR was surveyed using the four MOC parameters and their effect on the multiplication factor. To survey the four MOC parameters, each single MOC parameter was surveyed while all of the other MOC parameters were held constant. The four MOC parameters were surveyed for the reflective boundary condition and the vacuum boundary condition. The output plots for the homogeneous survey of the reflective and the vacuum boundary conditions are not shown because they are essentially the same as the boundary condition plots shown above except that they are scaled. However, in the case of the heterogeneous survey, they are much different than the boundary condition plots and will be shown in Chapter 5.

Homogeneous Survey Performed with Reflective Boundary Condition

The reflective boundary condition prevents any neutron leakage which is equivalent to simulating an infinite reactor. The CPU time was also recorded to assist in determining the usefulness of each calculation given its accuracy. For the reflective boundary condition, the value for K-inf = 0.19537 for all of the parameters that were changed throughout the survey. The CPU times varied between each parameter but the difference in CPU time was always less than a minute, so it was considered negligible.

Homogeneous Survey Performed with the Vacuum Boundary Condition

The vacuum boundary condition allows for the escape of all neutrons throughout the simulation. The K-effective term varies substantially more in the vacuum boundary condition than the reflective boundary condition. This makes sense because if the neutrons are not allowed to escape throughout the simulation, the K-inf will be unchanging. The proceeding discussion will show what those changes actually are.

Polar Angle Number and Scheme (Vacuum)

The corresponding table of K-eff. for the survey of the polar angles and number scheme can be seen in Table 4.1.

In this case, as in most of the cases for the homogeneous survey of the UUTR, the K-effective remained very similar. This is expected because of the use of the test material of the homogeneous model of the UUTR. This is also shown in Figure 4.5.

Table 4.1. Table of Polar Angle Scheme Survey for the Homogeneous UUTR (Vacuum B.C.).

Polar Angle Scheme	CPU Time	K - effective
1,1	843.0 s	0.13277
1,2	844.8 s	0.13277
2,1	678.2 s	0.134368
2,2	844.9 s	0.132932

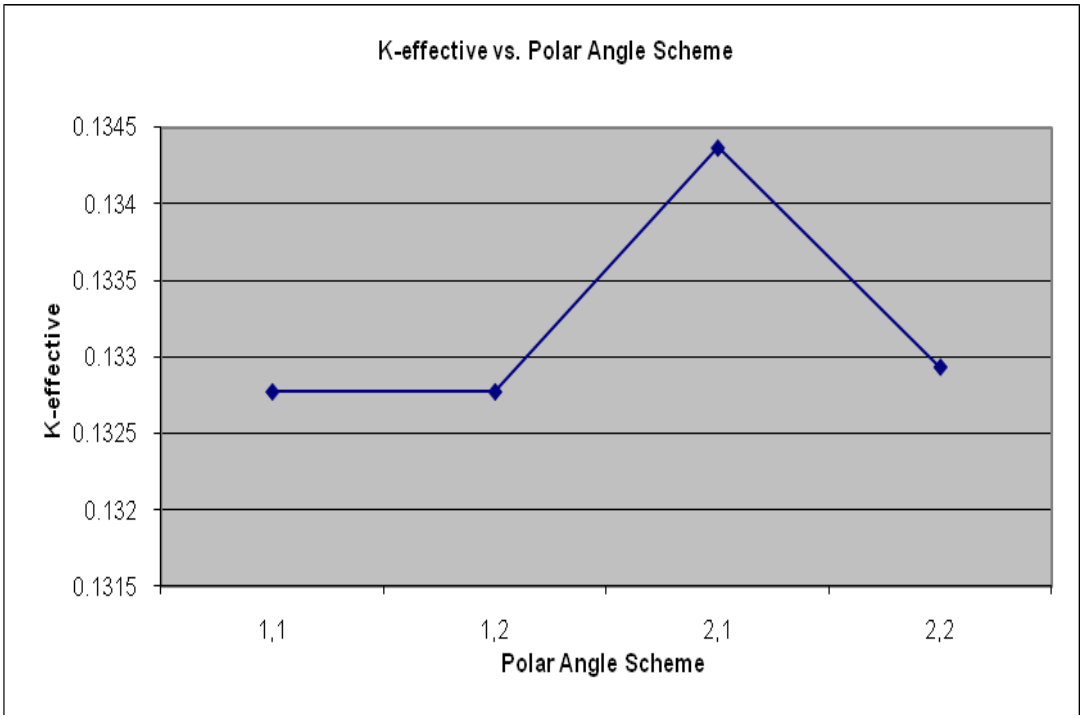


Figure 4.5. Plot of Polar Angle Scheme Survey for Homogeneous UUTR (Vacuum B.C.).

Azimuthal Angles (Vacuum B.C.)

The corresponding table of K-effective for the survey of the polar angles and number scheme can be seen in Table 4.2 and Figure 4.6, respectively.

In this case, as in most of the cases for the homogeneous survey of the UUTR, the K-effective remained very similar. This is expected because of the use of the test material of the homogeneous model of the UUTR. This is shown in Figure 4.6.

Ray Separation (Vacuum B.C.)

The corresponding table of K-effective for the survey of the polar angles and number scheme can be seen below in Table 4.3 and Figure 4.7, respectively.

In this case, as in most of the cases for the homogeneous survey of the UUTR, the K-effective remained the same until the threshold of 0.04 cm was reached. Although the computation time increased substantially at 0.02 cm with seemingly little additional benefit in the accuracy of K-effective, it will be shown in the heterogeneous survey of the UUTR that it is very valuable to use the 0.02 cm ray separation. This survey is shown in Figure 4.7.

Table 4.2. Table of Azimuthal Angle Survey of the Homogeneous UUTR (Vacuum B.C.).

Azimuthal Angle	CPU Time	K - effective
12	286.7 s	0.132718
20	463.5 s	0.132777
24	550.6 s	0.132759
32	748.5 s	0.132779
36	844.8 s	0.13277

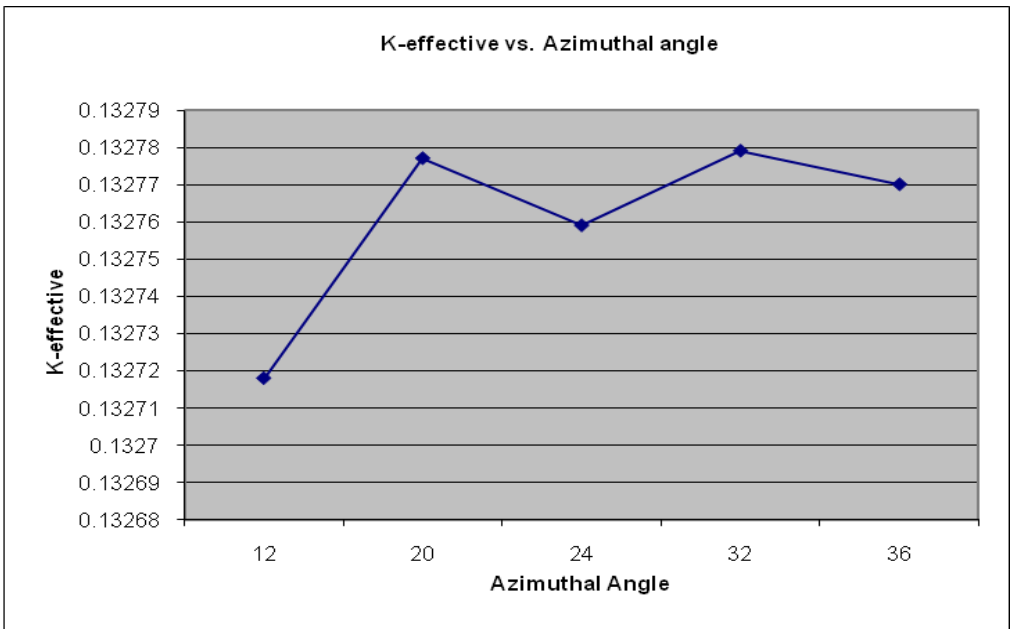


Figure 4.6. Plot of Azimuthal Angle Survey for the Homogeneous UUTR (Vacuum B.C).

Table 4.3. Table of Ray Separation Survey for the Homogeneous UUTR (Vacuum B.C.).

Ray Separation (cm)	CPU Time	K - effective
0.08	107.0 s	0.19537
0.06	145.0 s	0.19537
0.04	424.6 s	0.132776
0.02	844.8 s	0.13277

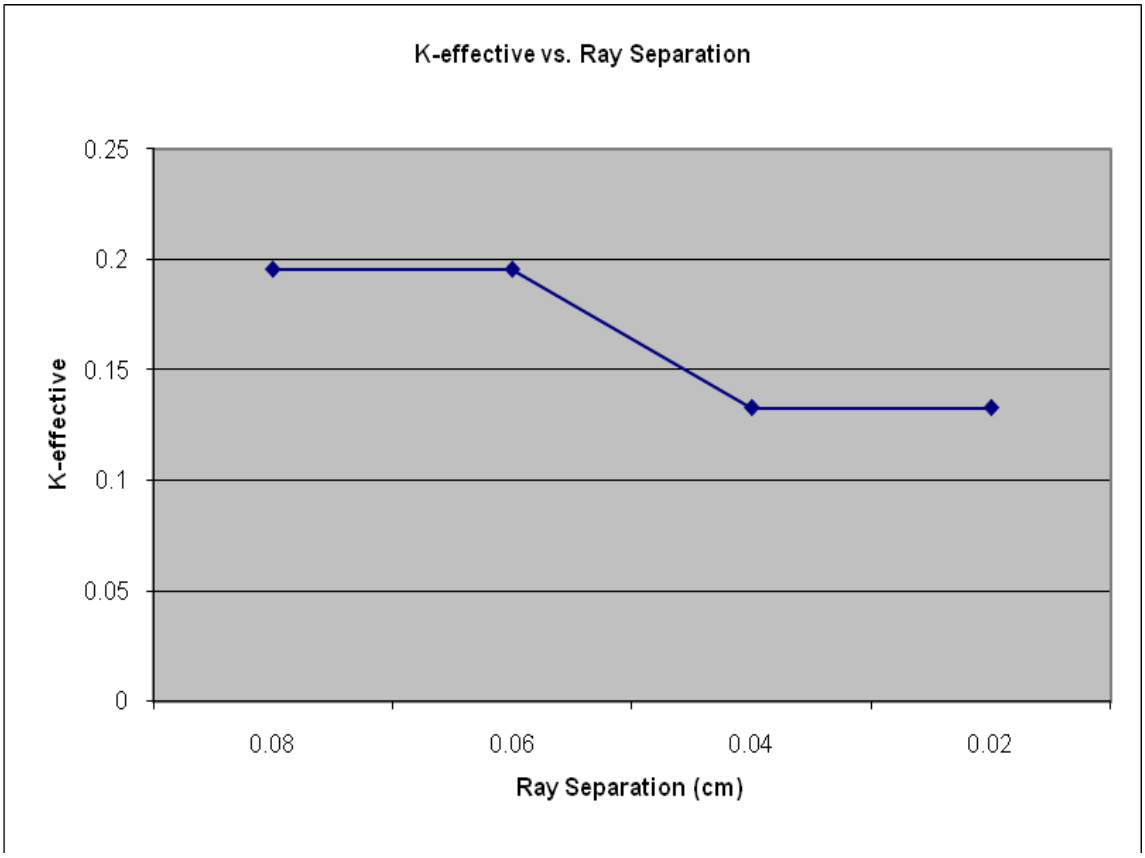


Figure 4.7. Plot of the Ray Separation Survey of the Homogeneous UUTR (Vacuum B.C.).

Number of Edges (Vacuum B.C.)

The corresponding table of K-effective for the survey of the polar angles and number scheme for the homogeneous TRIGA with the vacuum boundary condition can be seen in Table 4.4 and Figure 4.8, respectively.

In this case, the K-effective remained unchanged, which is expected because of the use of the test material of the homogeneous model of the UUTR, in combination with the number of edges per face. This is shown in Figure 4.8.

Comparison of Reflective versus Vacuum Boundary Condition for Homogenous UUTR

The values of K-inf. for the reflective boundary condition and the K-eff. for the vacuum boundary condition of the survey of the homogeneous UUTR only varied by approximately 0.07. This can be seen in Figures 4.9 through 4.12.

Table 4.4. Table of # of Edges Survey for Homogeneous UUTR (Vacuum B.C.).

# of Edges	CPU Time	K - effective
4	844.8 s	0.13277
12	905.6 s	0.13277
24	846.1 s	0.13277
36	842.6 s	0.13277

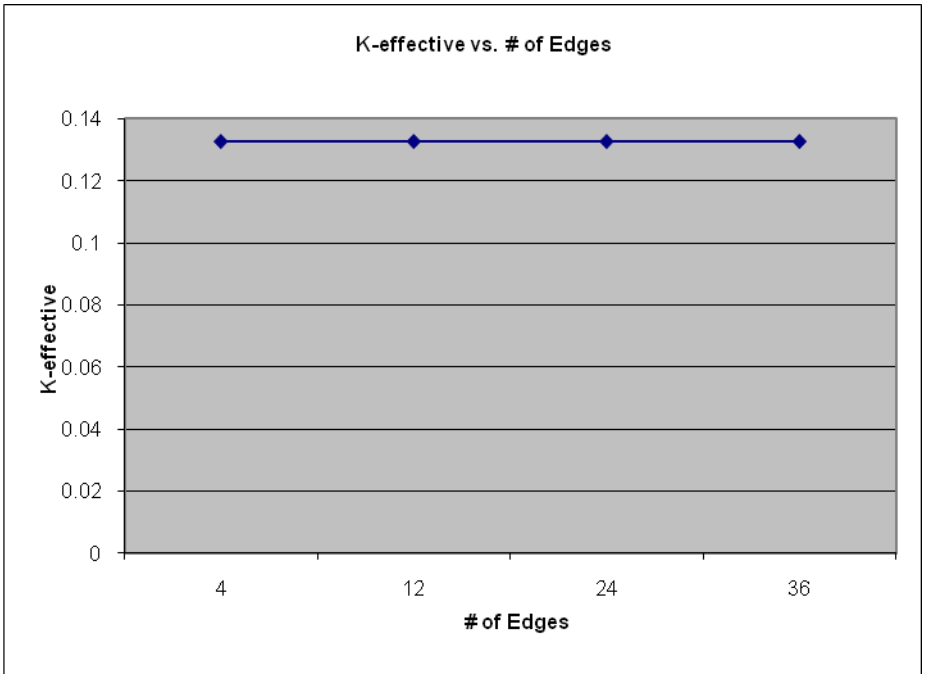


Figure 4.8. Plot of the # of Edges Survey of the Homogeneous UUTR (Vacuum B.C.)

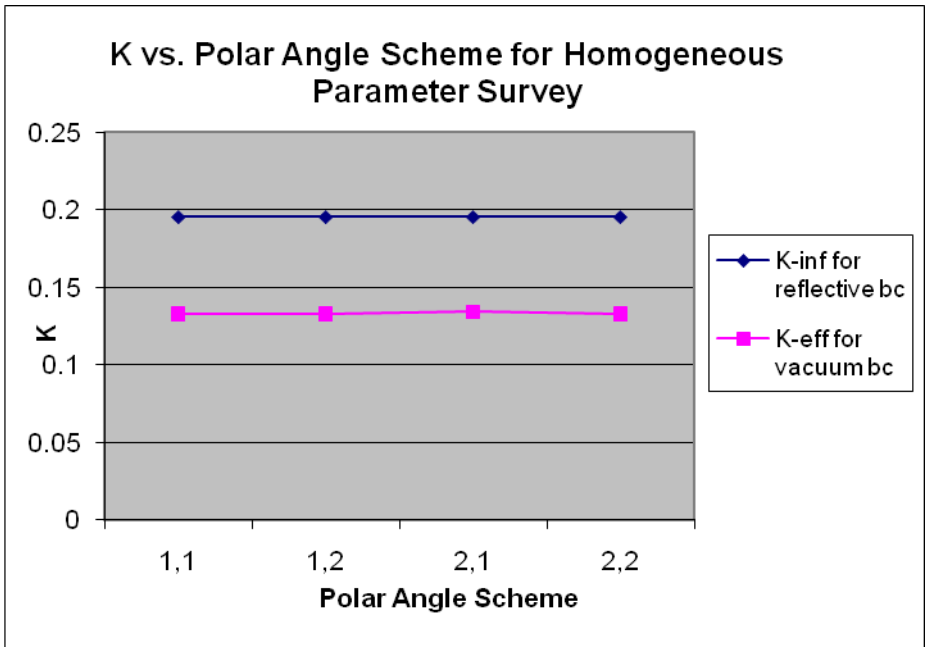


Figure 4.9. Comparison of K-inf. vs. K-eff. for the Reflective vs. Vacuum B.C. for Polar Angle Survey.

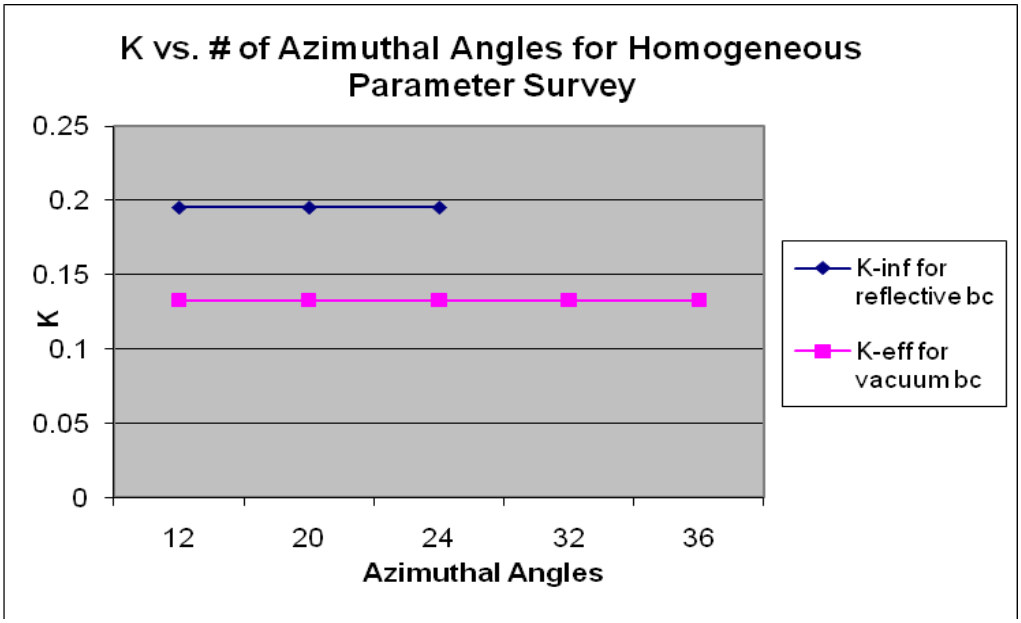


Figure 4.10. Comparison of K-inf. vs. K-eff. for the Reflective vs. Vacuum B.C. for Azimuthal Angle Survey.

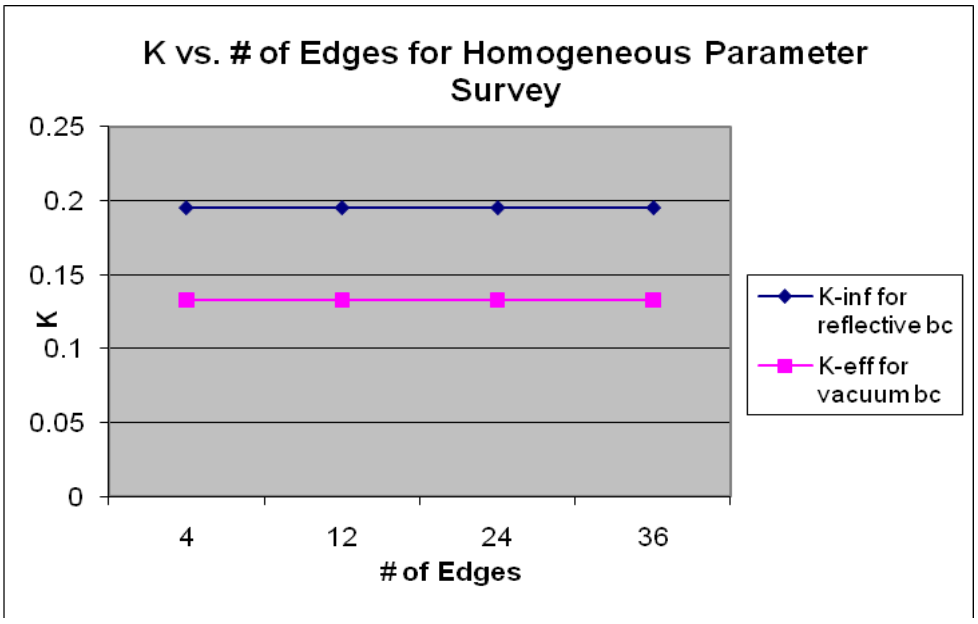


Figure 4.11. Comparison of K-inf. vs. K-eff. for the Reflective vs. Vacuum B.C. for the # of Edges Survey.

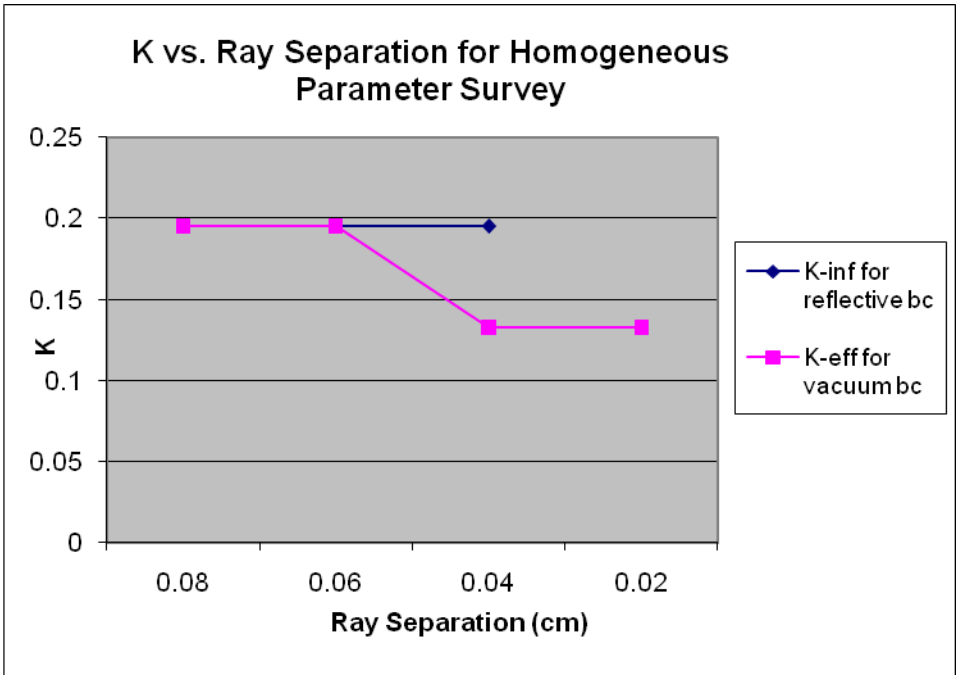


Figure 4.12. Comparison of K-inf. vs. K-eff. for the Reflective vs. Vacuum B.C. for the Ray Separation Survey.

CHAPTER V

RESULTS OF THE SURVEY OF THE HETEROGENEOUS UUTR

The UUTR was surveyed using the four MOC parameters to see the effect on the multiplication factor. To survey the four MOC parameters, each single MOC parameter was surveyed while all other MOC parameters were held constant. The four MOC parameters were surveyed for the reflective boundary condition and the vacuum boundary condition. For the heterogeneous condition, these values varied greatly.

Heterogeneous Survey Performed with Reflective Boundary Condition

As stated earlier, the reflective boundary condition prevents any leakage from occurring which changes the k-effective to k-infinite. All of the surveys in the reflective boundary condition produced results that, when shown as the simulated reactor, appeared very similar. To avoid redundancy in the report, the flux map of the UUTR will only be shown once for each of the five energy groups for the reflective boundary condition. The absorption reaction rate, fission reaction rate, and fission source will also be shown just

once for the reflective boundary condition. These plots can be seen in Figures 5.1 and 5.2, respectively.

a)

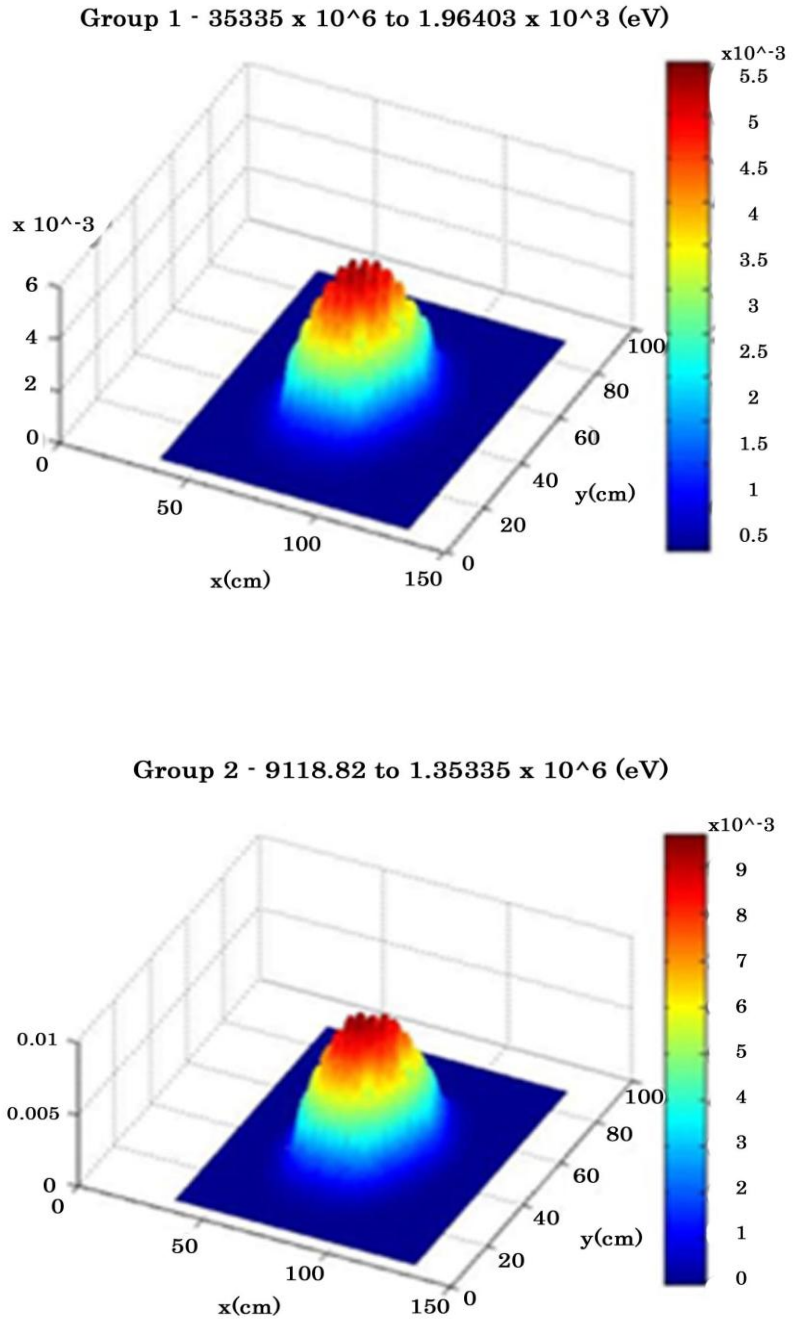
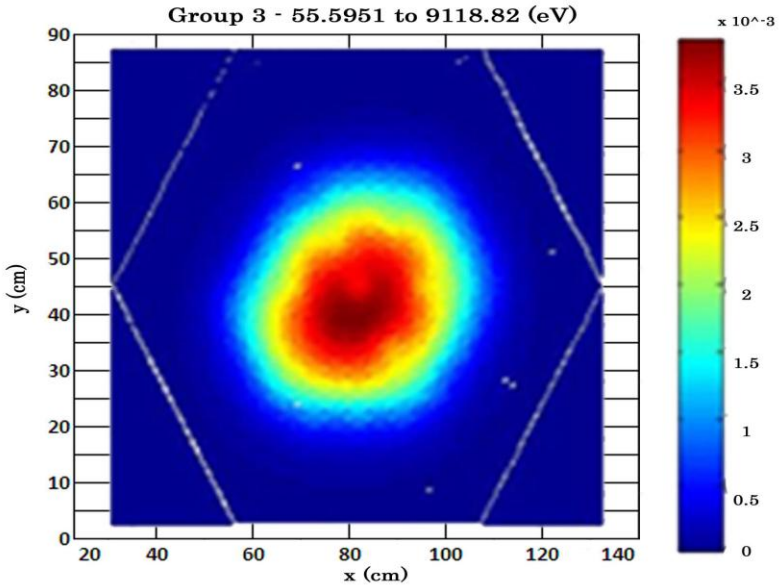


Figure 5.1. Plots of the 5 Energy Groups of the UUTR Heterogeneous for the Reflective Boundary Condition.

b)



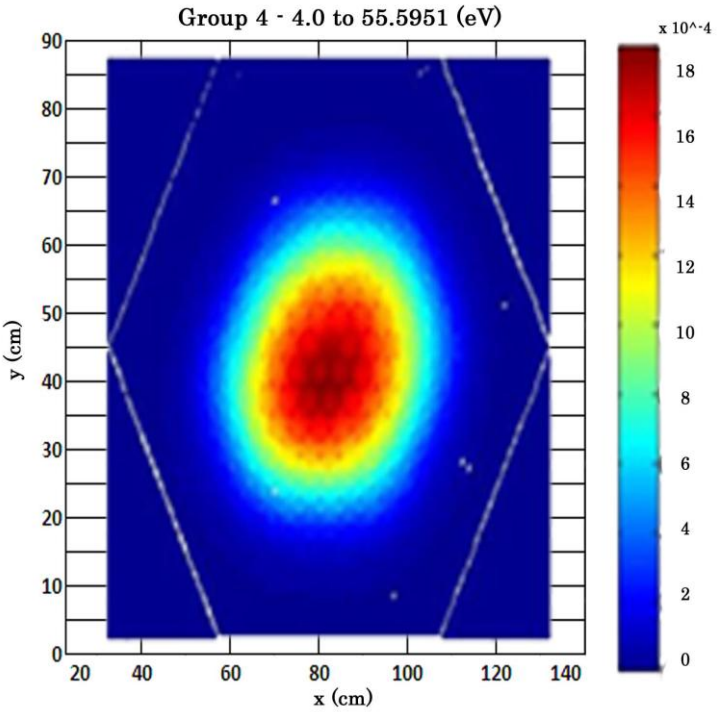


Figure 5.1. Continued

c)

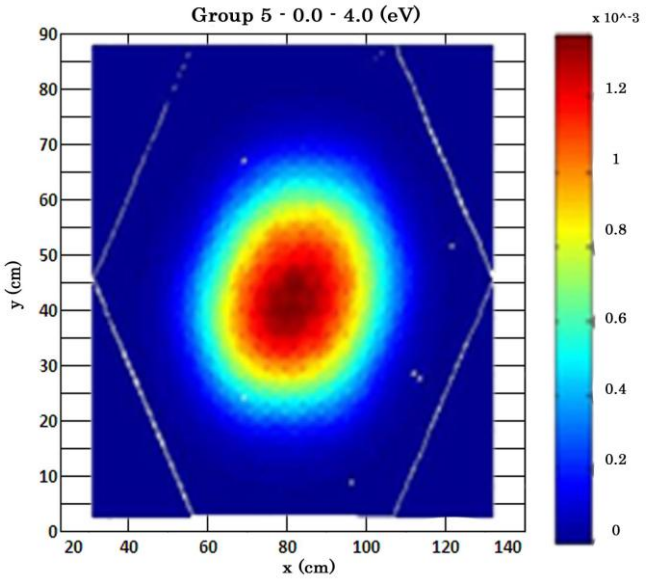


Figure 5.1. Continued

a)

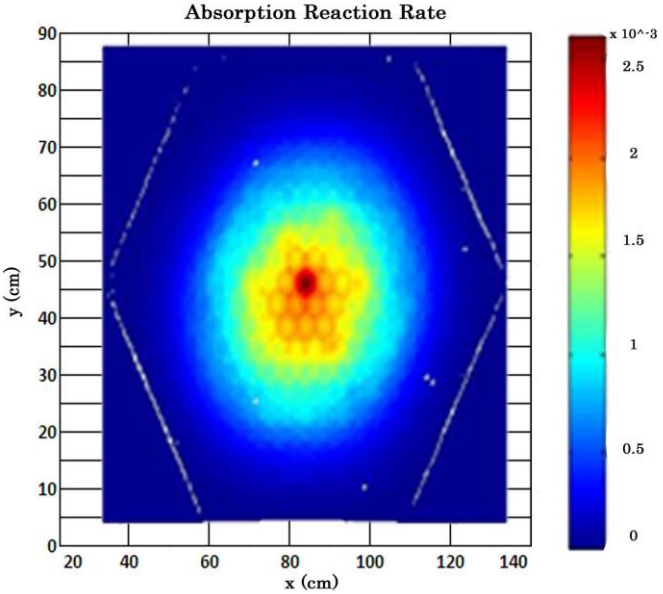


Figure 5.2. The Absorption Reaction Rate, The Fission Reaction Rate, and Fission Source for the UUTR Heterogeneous Reflective Boundary Condition.

b)

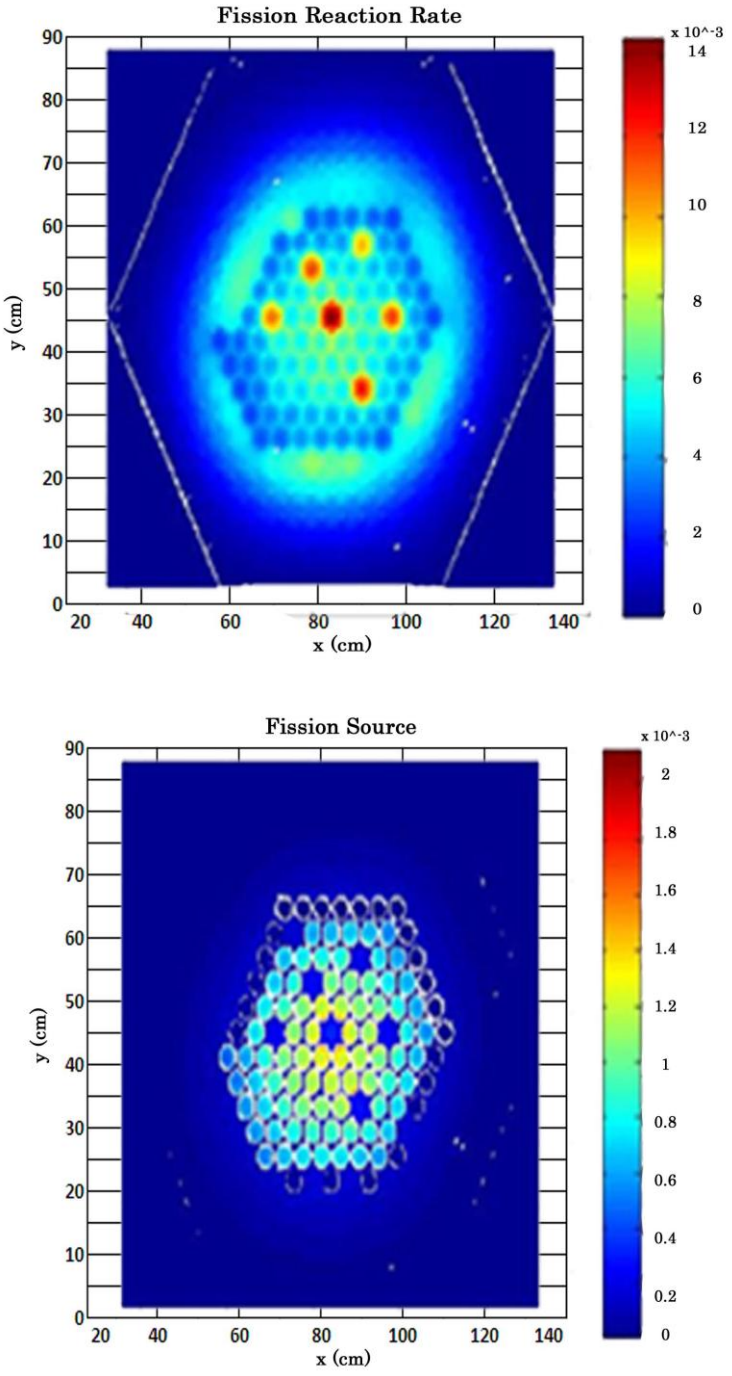


Figure 5.2. Continued

Heterogeneous Polar Angle Number and Scheme Survey (Reflective)

For the polar angle and number scheme, there are two options: the "Leonard and McDaniel" (LM) method and the "CACTUS" method. For the LM method, it was found that K-infinite was the same value regardless of the number of angles. This can be seen in Table 5.1 and the corresponding plot in Figure 5.3.

For the CACTUS method of the polar angle scheme, K-infinite varied between each number of polar angles used in the analysis but converged to a particular value as the number of polar angles increased. This is shown in Figure 5.4 and Table 5.2, respectively.

Table 5.1. K-Infinite for the LM Polar Angle Scheme for Heterogeneous UUTR with Reflective B.C.

Polar Angle Scheme (Leonard and McDaniel)	CPU Time	K - Infinite
1,1	6720.1 s	1.14846
1,2	6727.1 s	1.14846

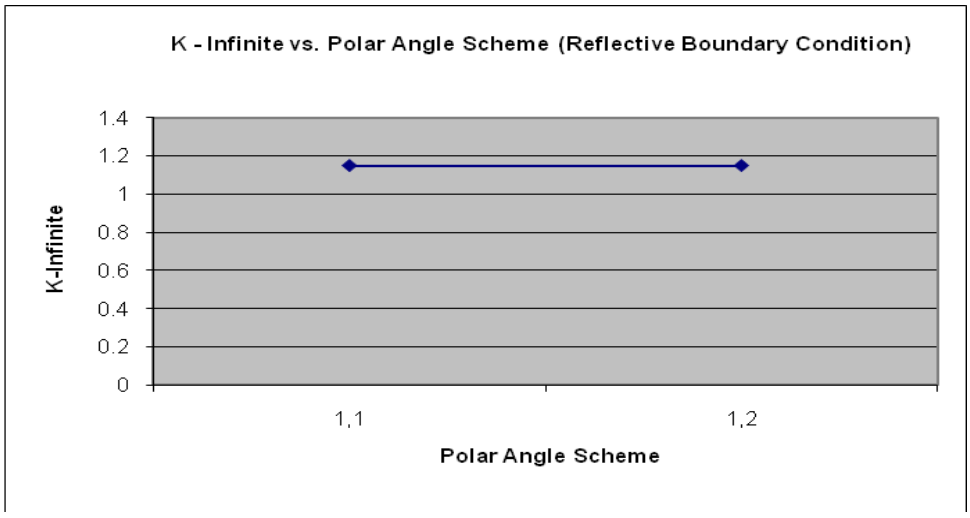


Figure 5.3. Plot of LM Polar Angle Scheme for Heterogeneous UUTR with Reflective Boundary Condition.

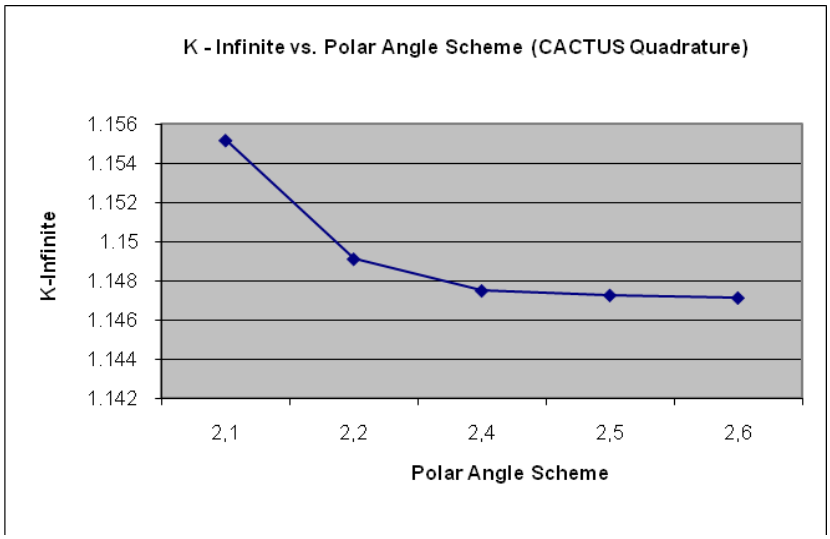


Figure 5.4. Plot of CACTUS Polar Angle Scheme for Heterogeneous UUTR with Reflective Boundary Condition.

Heterogeneous Azimuthal Survey (Reflective)

The table and plot of K-infinite for the heterogeneous UUTR with the reflective boundary condition during the azimuthal survey is shown in Figure 5.5 and Table 5.3, respectively.

Heterogeneous Ray Separation Survey (Reflective)

The table and plot of K-infinite for the heterogeneous UUTR with the reflective boundary condition during the azimuthal survey is shown in Figure 5.6 and Table 5.4, respectively.

Heterogeneous Number of Edges Survey (Reflective)

The table and plot of K-infinite for the heterogeneous UUTR with the reflective boundary condition during the number of edges survey is shown in Figure 5.7 and Table 5.5, respectivly.

Table 5.2. K-infinite for CACTUS Polar Angle Scheme.

Polar Angle Scheme (CACTUS)	CPU Time	K-Infinite
2,1	5617.4 s	1.155171
2,2	6861.6 s	1.149121
2,4	10890.4 s	1.147507
2,5	14081.3 s	1.147275
2,6	14616.7 s	1.147143

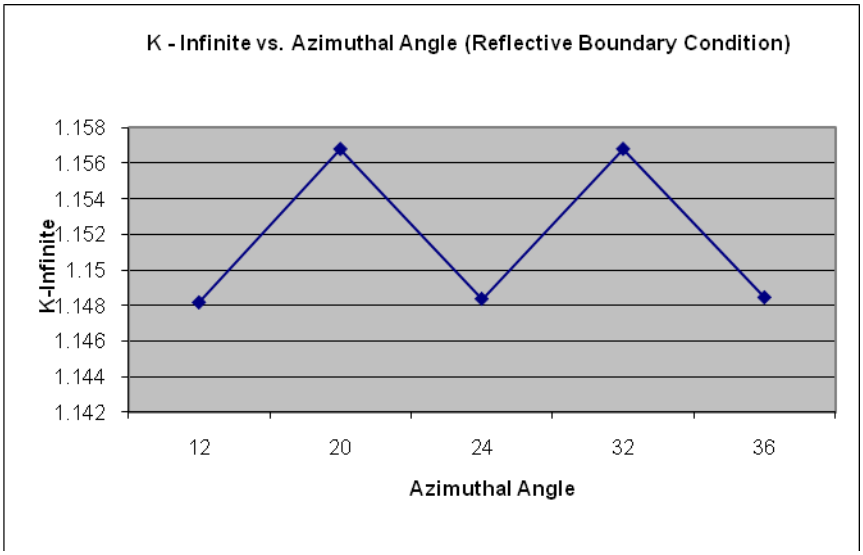


Figure 5.5. Plot of Azimuthal Angle Survey of the Heterogeneous UUTR with Reflective Boundary Condition.

Table 5.3. Table of Azimuthal Angle Survey for Heterogeneous UUTR with Reflective B.C.

Azimuthal Angle	CPU Time	K - Infinite
12	3018.8 s	1.148175
20	3507.0 s	1.156777
24	4450.5 s	1.148378
32	5645.5 s	1.156782
36	6727.1 s	1.14846

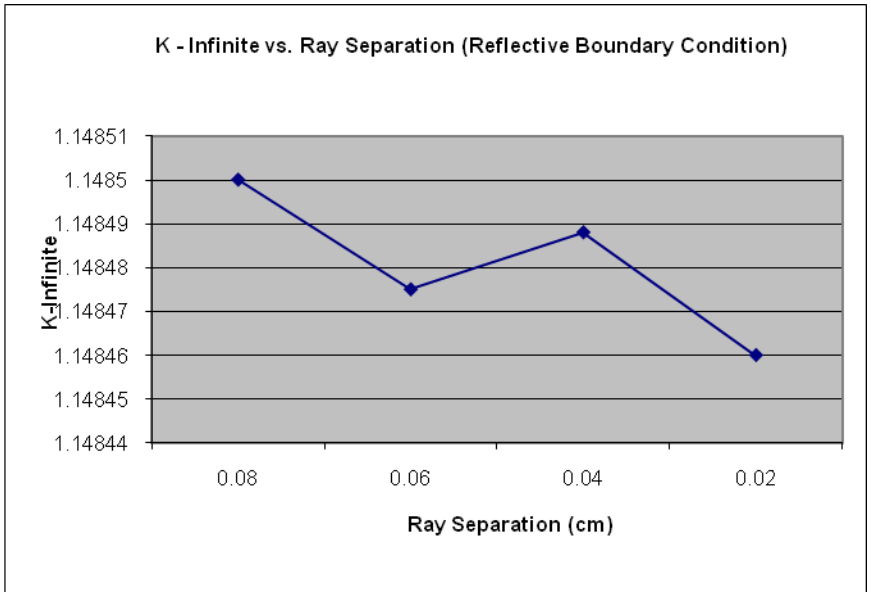


Figure 5.6. Plot of Ray Separation Survey of the Heterogeneous UUTR with Reflective Boundary Condition.

Table 5.4. Table of Ray Separation Survey for Heterogeneous UUTR with Reflective B.C.

Ray Separation (cm)	CPU Time	K - Infinite
0.08	1682.9 s	1.1485
0.06	2247.6 s	1.148475
0.04	3354.3 s	1.148488
0.02	6727.1 s	1.14846

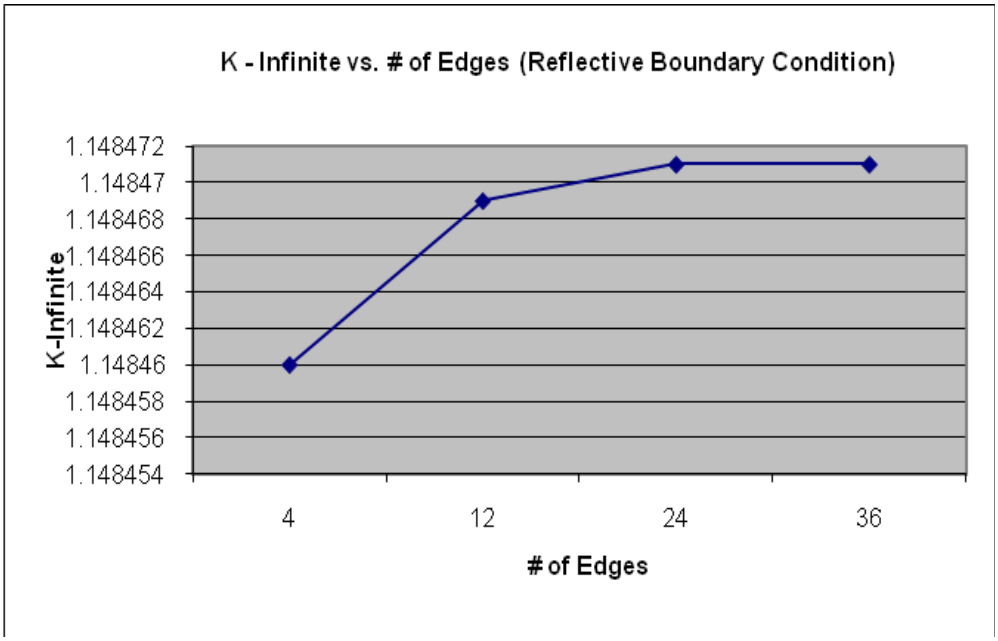


Figure 5.7. Plot of # of Edges Survey of the Heterogeneous UUTR with Reflective Boundary Condition.

Table 5.5. Table of # of Edges Survey for Heterogeneous UUTR with Reflective B.C.

# of Edges	CPU Time	K - Infinite
4	6727.1 s	1.14846
12	7142.9 s	1.148469
24	7222.7 s	1.148471
36	7080.5 s	1.148471

Heterogeneous Survey Performed with Vacuum Boundary Condition

As stated earlier, the vacuum boundary condition prevents any reflection from occurring on any of the boundaries which means the multiplication factor remains at k-effective. All of the surveys in the vacuum boundary condition produced results that, when shown as the simulated reactor, appeared very similar. To avoid redundancy in the report, the flux map of the UUTR will only be shown once for each of the five energy groups for the vacuum boundary condition just as it was for the reflective boundary condition. The absorption reaction rate, fission reaction rate, and fission source will also be shown just once for the reflective boundary condition just as before. These plots can be seen in Figures 5.8 and 5.9, respectively.

a)

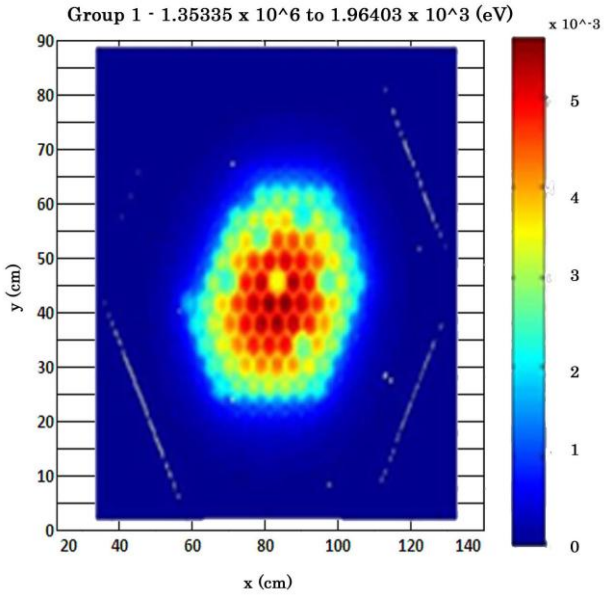


Figure 5.8. Plots of the 5 Energy Groups of the UUTR Heterogeneous for the Vacuum Boundary Condition.

b)

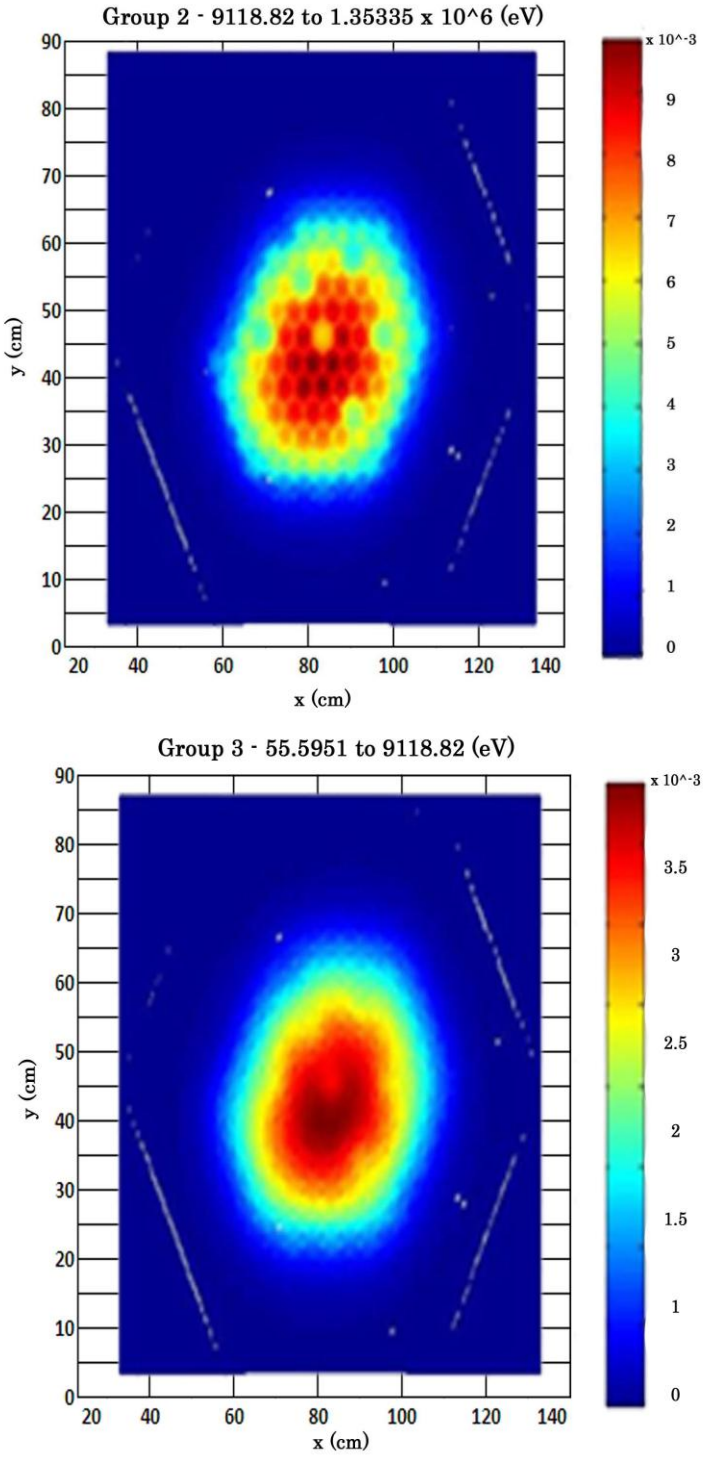


Figure 5.8. Continued.

c)

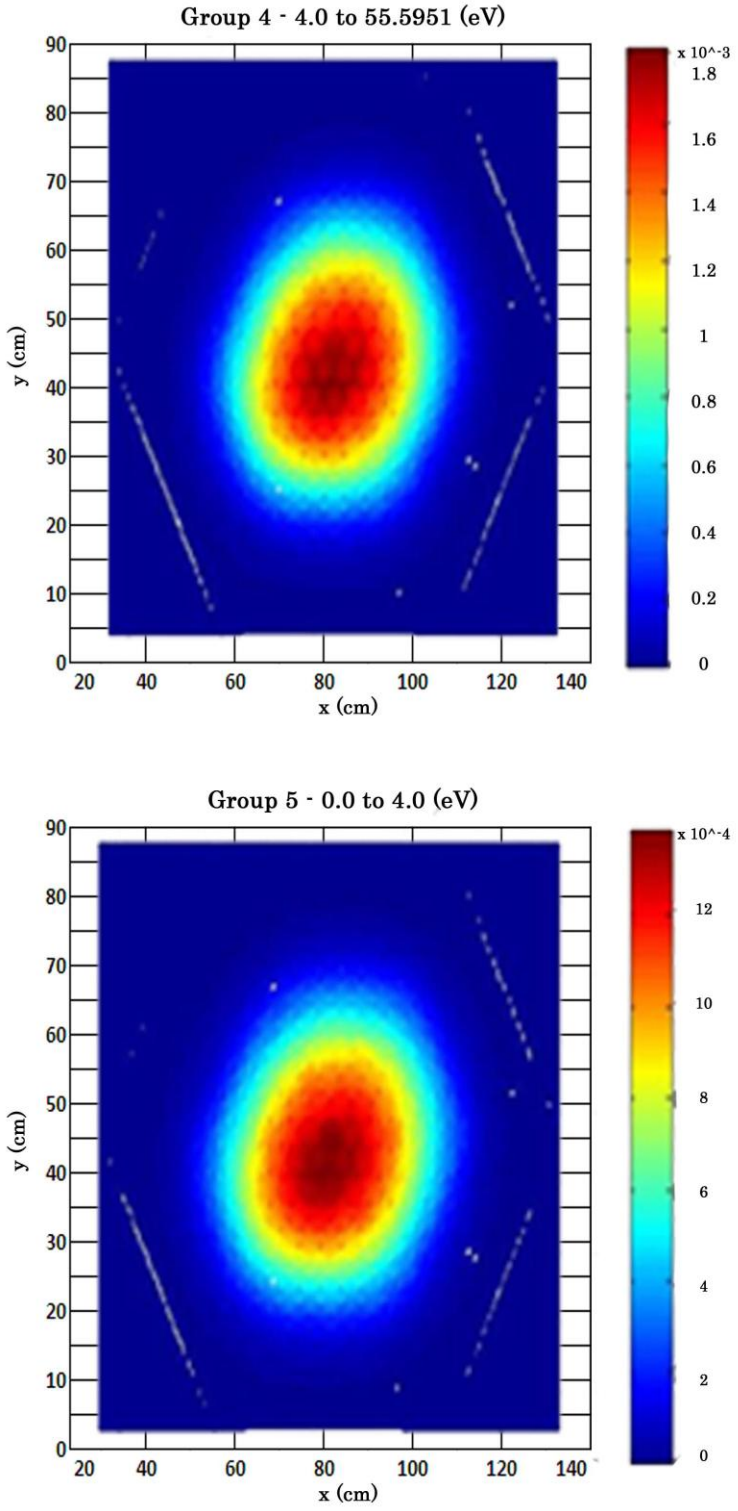
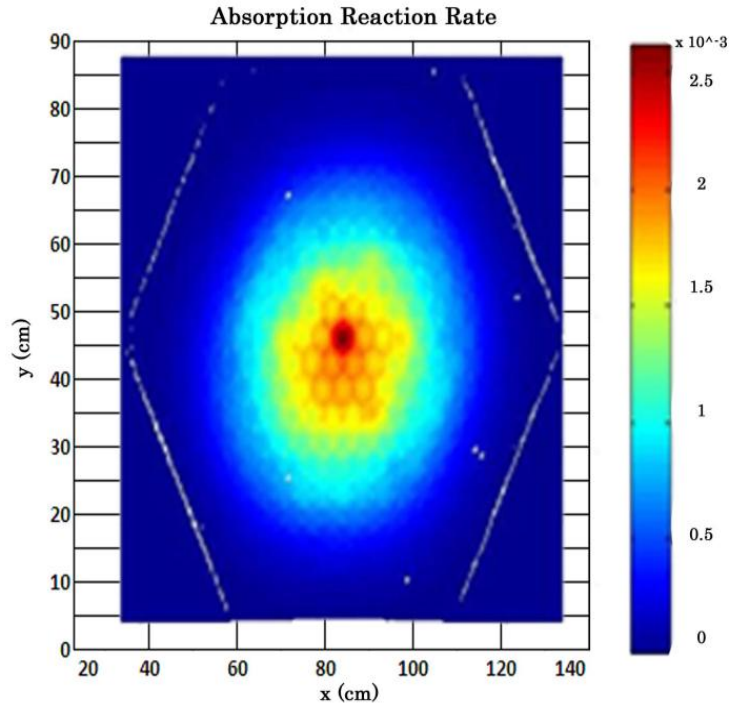


Figure 5.8. Continued.

a)



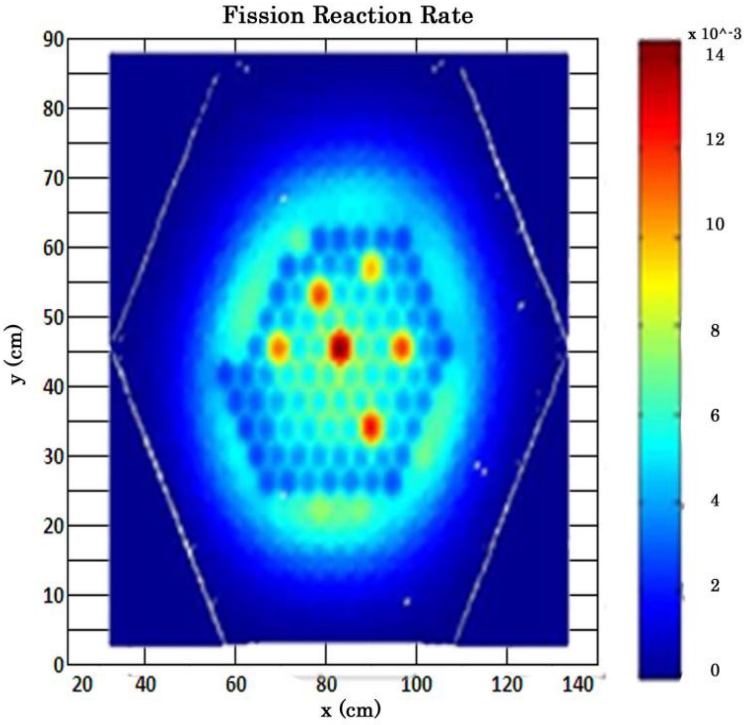


Figure 5.9. The Absorption Reaction Rate, The Fission Reaction Rate, and Fission Source for the UUTR Heterogeneous Reflective Boundary Condition.

b)

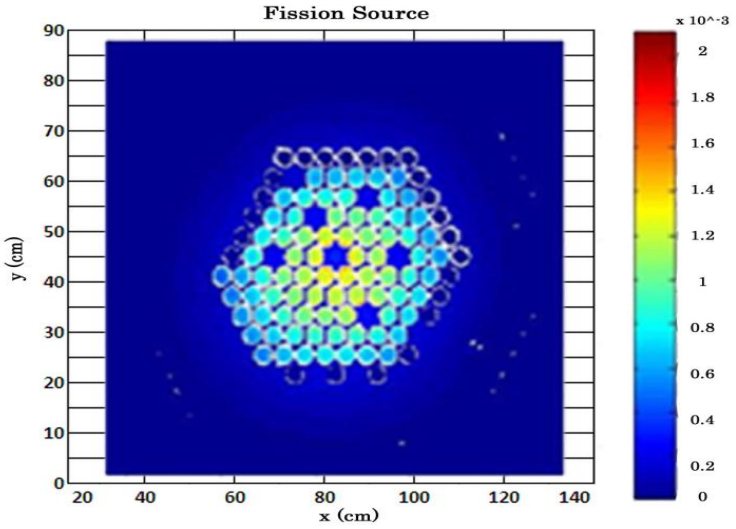


Figure 5.9. Continued

Heterogeneous Polar Angle Number and Scheme Survey (Vacuum)

For the polar angle and number scheme, there are two options for the vacuum boundary condition: the "Leonard and McDaniel" (LM) method and the "CACTUS" method. For the LM method it was found that K-infinite was the same value regardless of the number of angles. This can be seen in Table 5.6 and corresponding plot in Figure 5.10.

For the CACTUS method of the polar angle scheme, K-effective varied between each number of polar angles used in the analysis but converged to a particular value as the number of polar angles increased. Instead of converging up to some value as is the case for the reflective boundary condition, for the vacuum boundary condition, the K-effective converged down to some value. This is because of the leakage that is inherent to the vacuum boundary condition model. This is shown in Table 5.7 and Figure 5.11, respectively.

Table 5.6. Table of K-effective for the LM Polar Angle Scheme for Heterogeneous UUTR with Vacuum B.C..

Polar Angle Scheme (Leonard and McDaniel)	CPU Time	K - effective
1,1	5935.1 s	1.148272
1,2	5961.9 s	1.148272

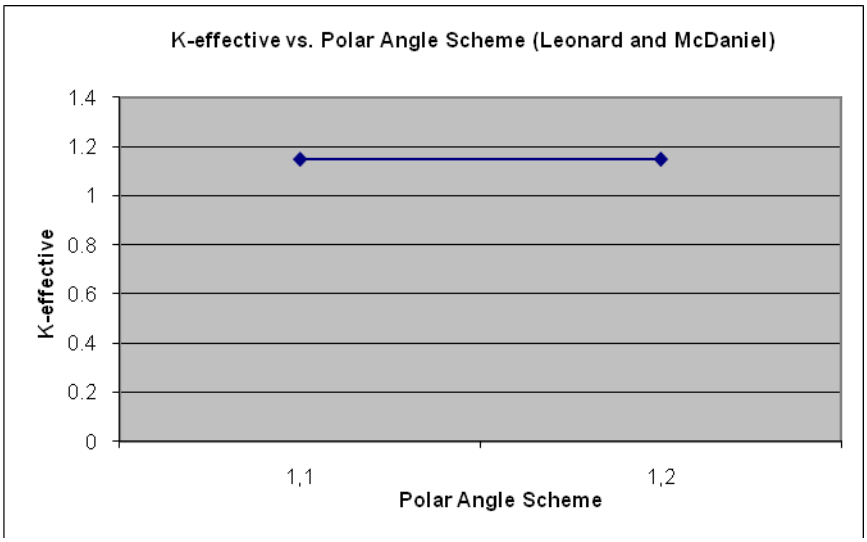


Figure 5.10. Plot of K-effective for the LM Polar Angle Scheme for Heterogeneous UUTR with Vacuum B.C..

Table 5.7. Table of K-effective for the CACTUS Polar Angle Scheme for Heterogeneous UUTR with Vacuum Boundary Condition.

Polar Angle Scheme (CACTUS)	CPU Time	K - effective
2,1	4909.6 s	1.155044
2,2	5961.2 s	1.148921
2,3	9359.0 s	1.147758
2,4	9582.6 s	1.14729
2,5	12600.6 s	1.147056
2,6	12895.0 s	1.146923

Heterogeneous Azimuthal Survey (Vacuum)

The table and plot of K-effective for the heterogeneous UUTR with the vacuum boundary condition during the azimuthal survey is shown in Table 5.8 and Figure 5.12, respectively.

Heterogeneous Ray Separation Survey (Vacuum)

The table and plot of K-effective for the heterogeneous UUTR with the vacuum boundary condition during the ray separation survey is shown in Table 5.9 and Figure 5.13, respectively.

Heterogeneous Number of Edges Survey (Vacuum)

The table and plot of K-effective for the heterogeneous UUTR with the vacuum boundary condition during the number of edges survey is shown in Table 5.10 and Figure 5.14.

Comparison of Reflective versus Vacuum Boundary Condition for Heterogeneous UUTR

The values of K-inf. for the reflective boundary condition and the K-eff. for the vacuum boundary condition of the survey of the homogeneous UUTR varied quite differently than the homogeneous survey of the UUTR, as seen in Figures 5.15 through 5.19.

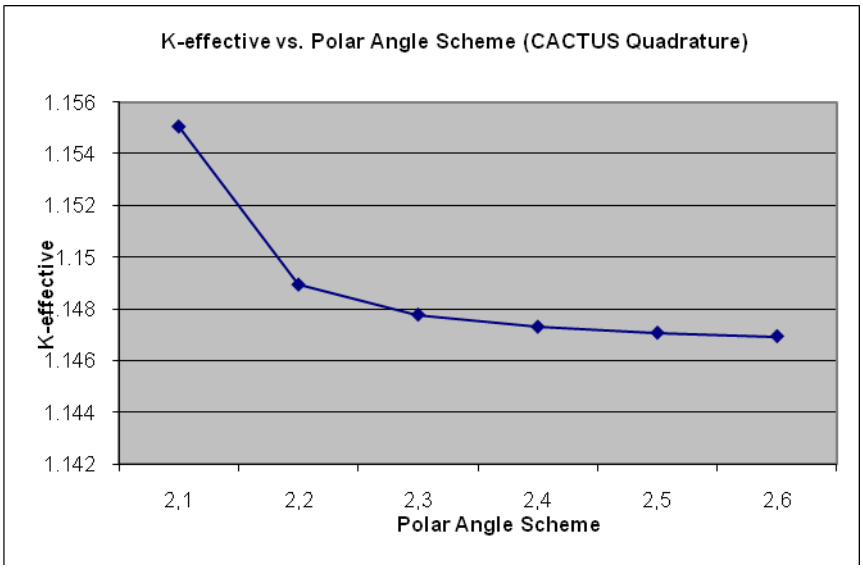


Figure 5.11. Plot of K-effective for the CACTUS Polar Angle Scheme for Heterogeneous UUTR with Vacuum Boundary Condition.

Table 5.8. Table of K-effective for the Azimuthal Angle Survey of the Heterogeneous UUTR with Vacuum B.C.

Azimuthal Angle	CPU Time	K - effective
12	1962.9 s	1.147985
20	3286.1 s	1.148304
24	4030.5 s	1.148189
32	5262.1 s	1.148315
36	5961.9 s	1.148272

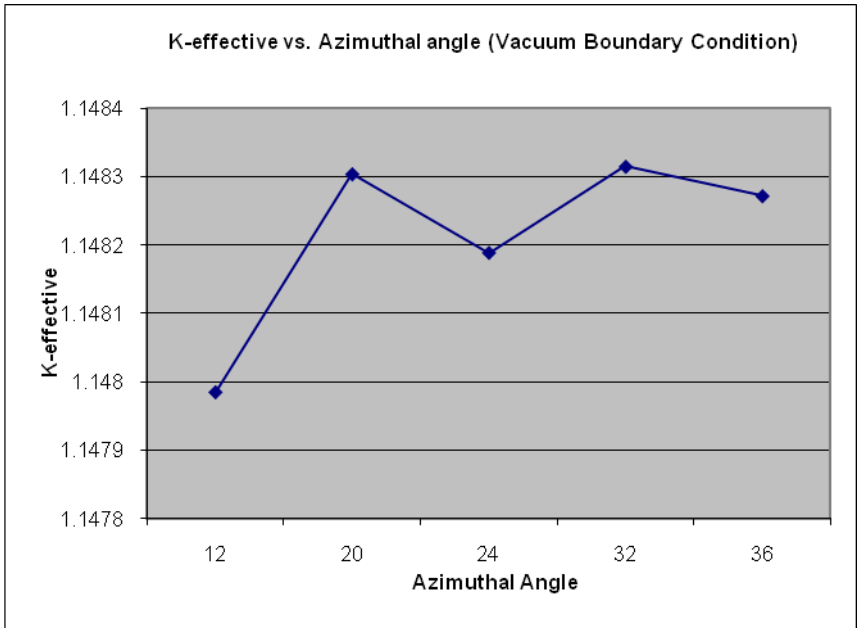


Figure 5.12. Plot of K-effective for the Azimuthal Angle Survey of the Heterogeneous UUTR with Vacuum B.C.

Table 5.9. Table of K-effective for the Ray Separation Survey of the Heterogeneous UUTR with Vacuum B.C..

Ray Separation (cm)	CPU Time	K - effective
0.08	1491.9 s	1.148312
0.06	2713.8 s	1.148286
0.04	2980.5 s	1.148299
0.02	5961.9 s	1.148272

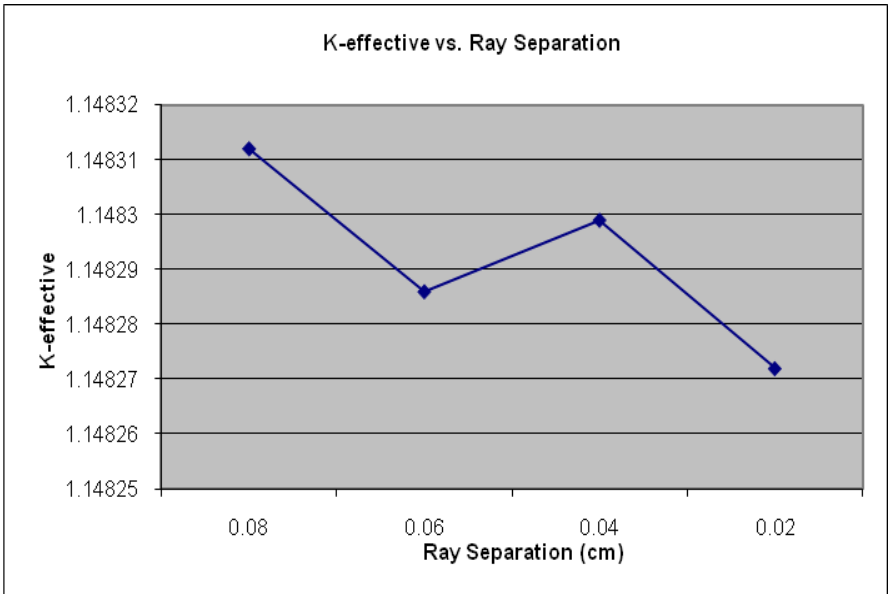


Figure 5.13. Plot of K-effective for the Ray Separation Survey of the Heterogeneous UUTR with Vacuum B.C..

Table 5.10. Table of K-effective for the # of Edges Survey of the Heterogeneous UUTR with Vacuum B.C..

# of Edges	CPU Time	K - effective
4	5961.9 s	1.148272
12	5939.9 s	1.148272
24	5961.7 s	1.148272
36	5979.6 s	1.148272

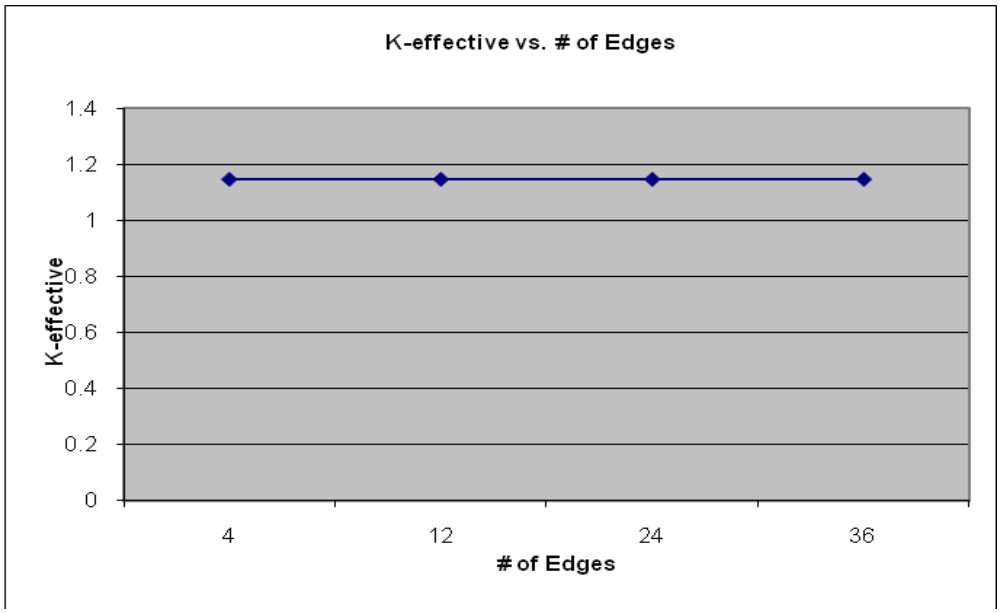


Figure 5.14. Plot of K-effective for the # of Edges Survey of the Heterogeneous UUTR with Vacuum B.C..

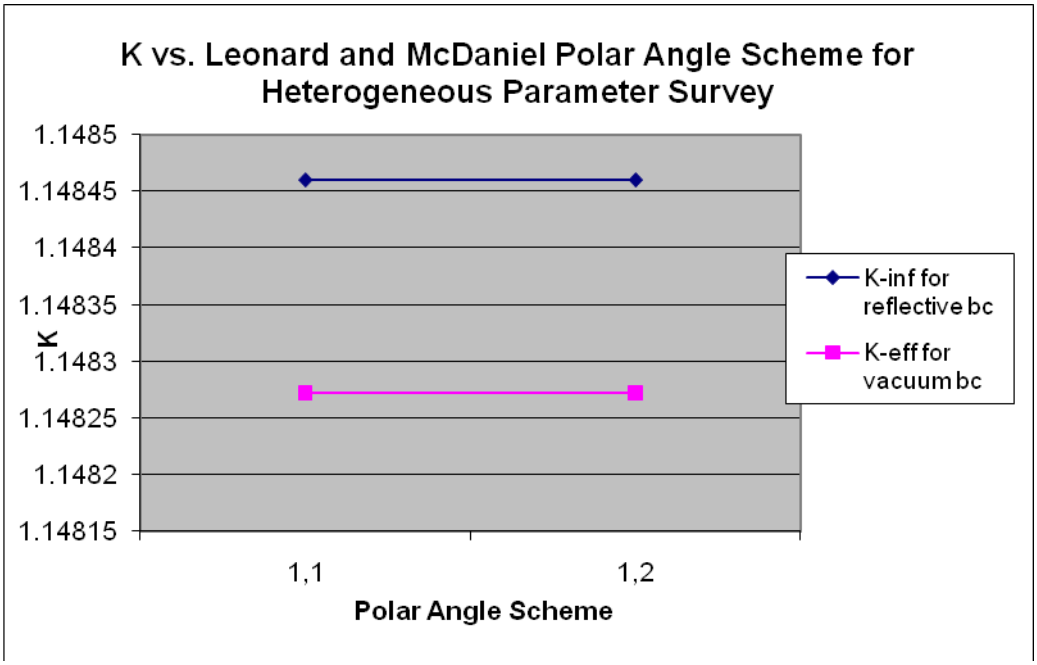


Figure 5.15. Comparison of K-inf. vs. K-eff. for the Heterogeneous Reflective vs. Vacuum B.C. for LM Polar Angle Survey.

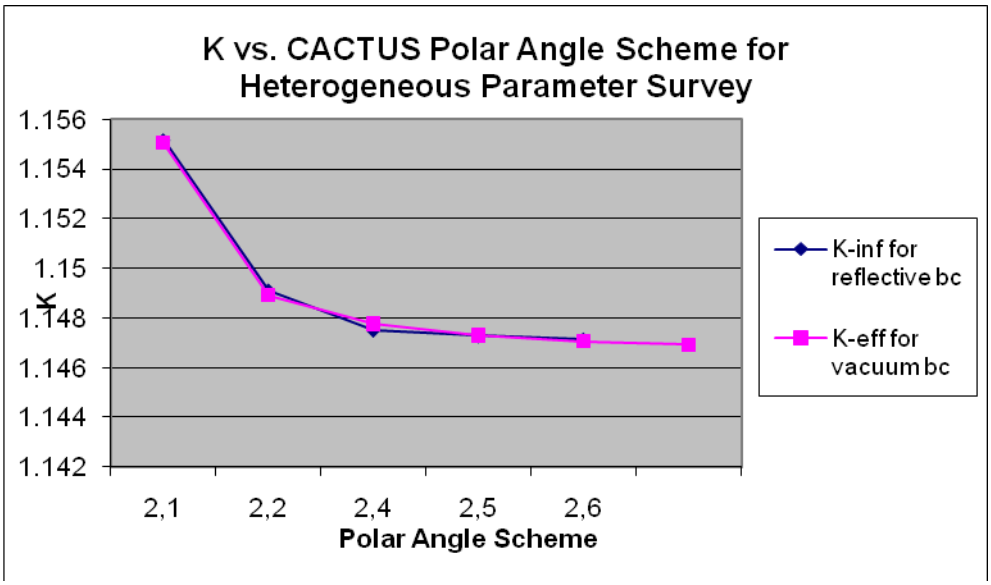


Figure 5.16. Comparison of K-inf. vs. K-eff. for the Heterogeneous Reflective vs. Vacuum B.C. for CACTUS Polar Angle Survey.

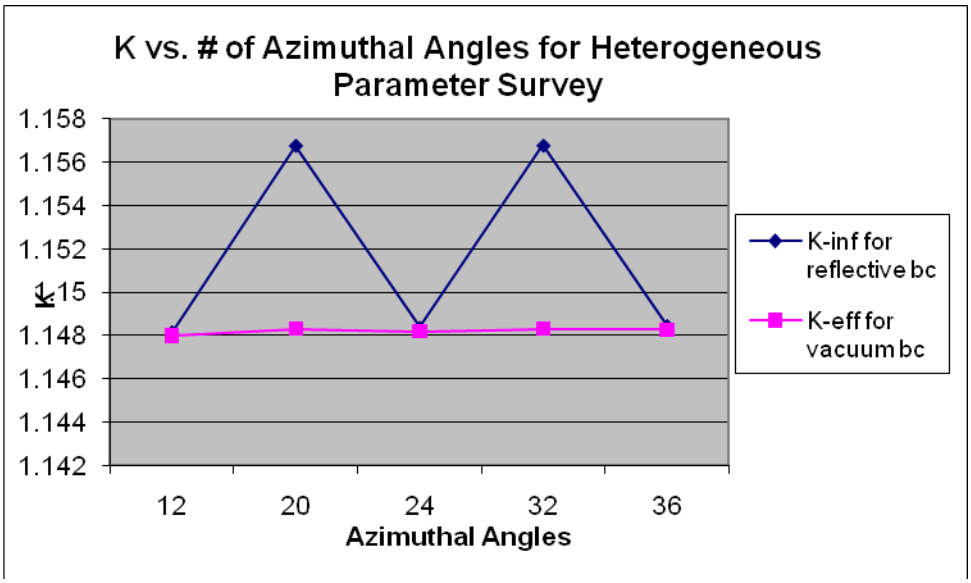


Figure 5.17. Comparison of K-inf. vs. K-eff. for the Heterogeneous Reflective vs. Vacuum B.C. Azimuthal Angle Survey.

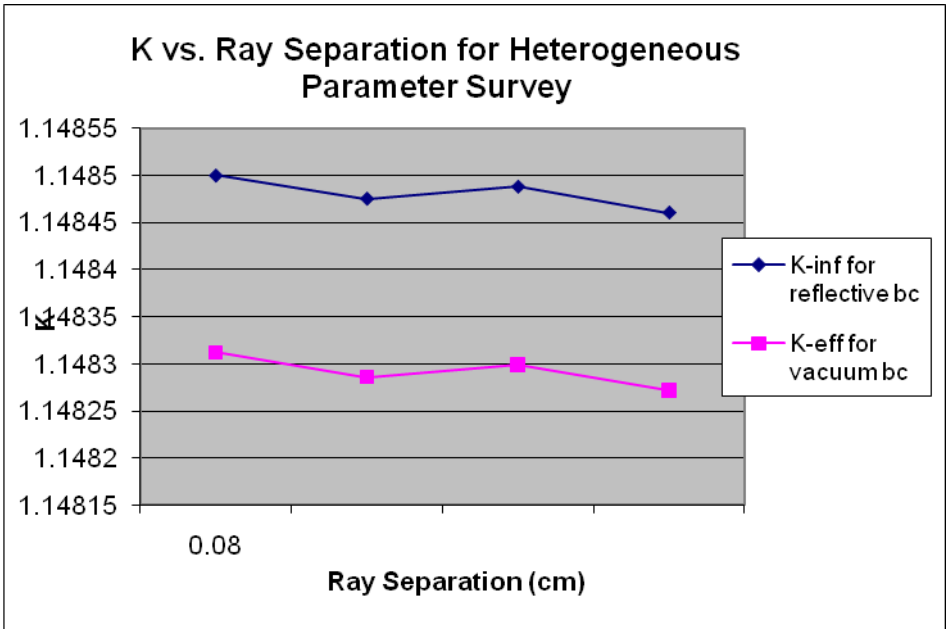


Figure 5.18. Comparison of K-inf. vs. K-eff. for the Heterogeneous Reflective vs. Vacuum B.C. Ray Separation Survey.

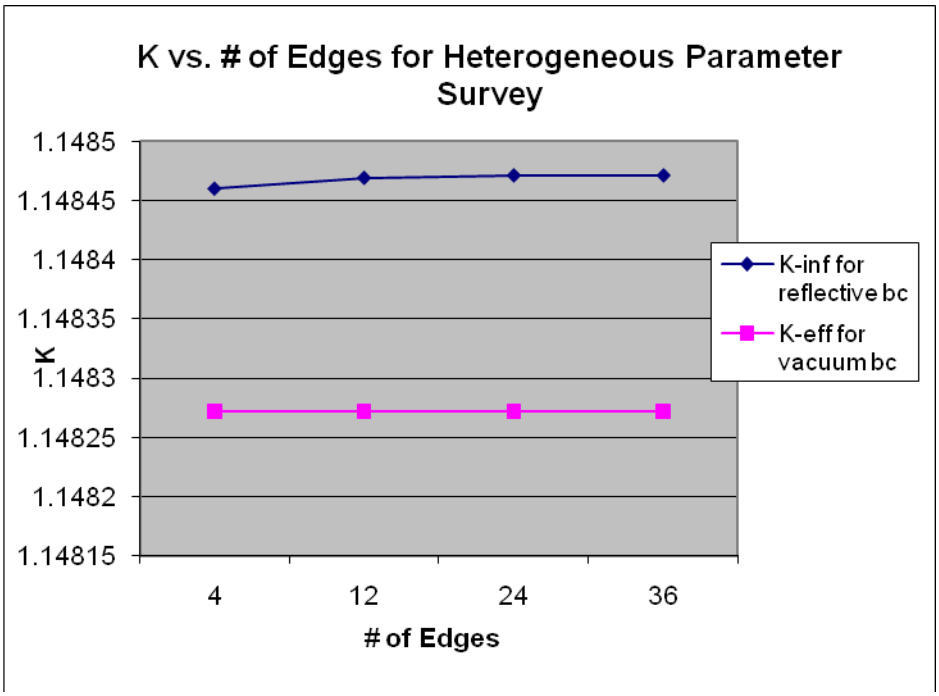


Figure 5.19. Comparison of K-inf. vs. K-eff. for the Heterogeneous Reflective vs. Vacuum B.C. # of Edges Survey.

CHAPTER VI

CONCLUSIONS

Summary

For the homogeneous simulation of the UUTR, the best k-inf. or k-eff. were still under 0.2. But this was expected because of the test material that was used. The k-inf. remained unchained throughout the entire survey of the four MOC parameters, of which were the polar angle scheme and number, the number of azimuthal angles, the ray separation, and the number of edges per side. For the vacuum boundary condition, the k-eff. did vary as the MOC parameters were changed, but it was minimal. There was an average difference of 0.06 between the k-inf. and k-eff. values.

For the heterogeneous simulation of the UUTR, the data for the cross-sections of the test material were replaced with the data for the cross-sections of the materials that were actually used in the UUTR, which was provided by a former graduate student at the University of Utah. This increased the accuracy of the actual k-eff. substantially while running in real-time. There was also a more prevalent change in the k-inf. and k-eff. as the MOC parameters were surveyed. This was a very valuable experience in learning AGENT.

The final MOC parameters that were used for the polar angle scheme and number were the Leanard and McDaniel scheme with 2 polar angles. The optimized number of

azimuthal angles was 36. The final ray separation parameter that was used was 0.02 cm. Lastly, the final number of edges per segment that was used in the input file was 4. These final values were found to give the optimal k-effective and k-infinite, with regards to accuracy, computation time, and stability of calculation.

Future Work

The first step in improving the simulation would be to run the simulation of the UUTR in 3-D for the heterogeneous case of the UUTR. It may also be prudent to run additional calculations for the appropriate material cross-sections to discover any possible discrepencies.

APPENDIX A

DEFINITION OF TERMS

Multiplication Factor (k)

The multiplication factor is a term used in the management of the neutron population in a reactor. The multiplication factor is very important because the following equations (1a through 1c) are true with regards to a reactor.

$$k < 1 \quad subcritical \tag{1a}$$

$$k = 1 \quad critical \tag{1b}$$

$$k > 1 \quad supercritical \tag{1c}$$

meaning that when k < 1, there are less neutrons produced then are lost, so the reactor is not self-sustaining. When k = 1, the same amount of neutrons that are produced are also lost, so the reactor is in steady-state. Finally, when k > 1, there are more neutrons being produced then there are losses, which occurs when powering up a reactor [5].

The multiplication factor can be shown in several ways. They are shown in equations 2a and 2b.

$$k = \frac{Number\ of\ Neutrons\ in\ the\ Current\ Generation}{Number\ of\ Neutrons\ in\ the\ Previous\ Generation} \quad (2a)$$

$$k = \frac{Rate\ of\ Neutron\ Production\ in\ Reactor}{Rate\ of\ Neutron\ Losses\ (absorption\ and\ leakage)\ in\ Reactor} \quad (2b)$$

A more formal way to write the multiplication factor is by using the four-factor formula and the six-factor formula. The four-factor formula assumes no leakage, which turns the k into k-infinite. The six-factor formula accounts for leakage. They are both described by equations 3a and 3b.

$$k_{\infty} = \eta f p \epsilon \quad (3a)$$

$$k = \eta f p \epsilon P_{FNL} P_{TNL} \quad (3b)$$

Where $\eta = \frac{Number\ of\ fission\ neutrons\ produced}{Number\ of\ neutrons\ absorbed\ in\ fuel}$

$$f = themal\ utilization = \frac{\Sigma_f^F}{\Sigma_a^F}$$

$$\epsilon = fast\ fission\ factor = \frac{Total\ number\ of\ fission\ neutrons\ (fast\ and\ thermal)}{number\ of\ fission\ nuetrons\ from\ thermal\ fission}$$

$$p = resonance\ escape\ probability$$

$$= \frac{Number\ of\ fission\ neutrons\ that\ slow\ down\ from\ fission\ to\ thermal\ energies}{Total\ number\ of\ fission\ nuetrons}$$

$$P_{FNL} = Fast\ non-leakage = Probability\ that\ fast\ neutrons\ will\ not\ leak\ out$$

$$P_{TNL} = thermal\ non-leakage = Probability\ that\ thermal\ neutron\ will\ not\ leak\ out$$

REFERENCES

[1] Nuclear Fuels Group. http://www.ga-esi.com/triga/ (accessed Jan 10, 2011).

[2] Xiao, S. *3D Reactor Core Modeling: Based on Advance AGENT Methodology*. Purdue University: Indiana, 2006.

[3] Jevremovic, T.; Xiao, S.; Satvat, N.; Gert, G.; Peng, Y.; Hursin, M.; Burns, A.; Xue, Y. *Welcome to 2D Single-AGENT Reactor Physics Code: AGENT 2D Single Students Manual*. Purdue University: Indiana, 2006.

[4] Jevremovic, T. *ANEMONA General Geometry Neutron Transport By MOC: Theory and Code Usage.* Nuclear Fuel Industries, Ltd. BWR Fuel Division: Tokyo, 1999.

[5] Duderstadt, J.; Hamilton, J. *Nuclear Reactor Analysis*. Ann Arbor: University of Michagan John Wiley and Sons: Michagan, 1976.

[6] Jevremovic, T. *ANEMONA: Multi-Assembly Multipgroup General Geometry Method of Characteristics for Neutral Particle Transport Modeling: ANEMONA Code System*. Purdue University: Indiana, 2000.

[7] Satvat, N. Agent Code Front-End/Back-End Coupling and Virtual Reactor Development. Masters Thesis. Purdue University: Indiana, 2006.

[8] Satvat, N. Advancing the Agent Methodology to Include the Higher Order of Neutron Anisotropy With Accelerated Solutions. Ph.D. Thesis. Purdue University: Indiana, 2009.

[9] Gert, G. Neutron Transport in Boubly Heterogeneous Media of High Temperature Reactors. Ph.D. Thesis. Purdue University: Indiana, 2007.

[10] Xiao, S. 3D Reactor Core Modeling Based on Advanced Agent Methodology. Masters Thesis. Purdue University: Indiana, 2006.

[11] Xiao, S. *R-functions and Their Application in AGENT Solid Modeler. NEGE*. Purdue University: Indiana, 2005.

[12] Glasstone, S.; Sesonske, A. *Nuclear Reactor Engineering: Third Edition.* XanEdu Publishing: Michigan, 2010.

[13] Lamarsh, J. R.; Baratta, A. J. *Introduction to Nuclear Engineering: Third Edition*. Prentice Hall Inc: New Jersey, 2001.

[14] Jevremovic, T. *Nuclear Principles in Engineering: Second Edition*. Springer Science Business Media, LLC: Indiana, 2009.

[15] Shultis, J. K. Faw, R. E. *Fundamentals of Nuclear Science and Engineering: Second Edition*. Taylor & Francis Group: Florida, 2008.